

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

Rodrigo Rodrigues Pires de Mello

**PROPOSTA DE UM PROTOCOLO DE NEGOCIAÇÃO
EM SISTEMAS MULTIAGENTE PARA
AGENDAMENTO AUTOMÁTICO DE REUNIÕES**

Florianópolis

2016

Rodrigo Rodrigues Pires de Mello

**PROPOSTA DE UM PROTOCOLO DE NEGOCIAÇÃO
EM SISTEMAS MULTIAGENTE PARA
AGENDAMENTO AUTOMÁTICO DE REUNIÕES**

Trabalho de conclusão de curso apresentado como parte dos requisitos para a obtenção do Grau de Bacharel em ciências da computação.

Orientador: Prof. Dr. Ricardo Azambuja Silveira

Coorientador: Msc. Thiago Ângelo Gelaim

Florianópolis

2016

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Pires de Mello, Rodrigo Rodrigues

Proposta de um protocolo de negociação em sistemas multiagente para agendamento automático de reuniões / Rodrigo Rodrigues Pires de Mello ; orientador, Ricardo Azambuja Silveira ; coorientador, Thiago Ângelo Gelaim. - Florianópolis, SC, 2016.

97 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico. Graduação em Ciências da Computação.

Inclui referências

1. Ciências da Computação. 2. Inteligência Artificial. 3. Agentes. 4. Negociação. 5. Agendamento automático de reuniões. I. Silveira, Ricardo Azambuja. II. Gelaim, Thiago Ângelo. III. Universidade Federal de Santa Catarina. Graduação em Ciências da Computação. IV. Título.

Rodrigo Rodrigues Pires de Mello

**PROPOSTA DE UM PROTOCOLO DE NEGOCIAÇÃO
EM SISTEMAS MULTIAGENTE PARA
AGENDAMENTO AUTOMÁTICO DE REUNIÕES**

Trabalho de conclusão de curso apresentado como parte dos requisitos para a obtenção do grau de “Bacharel em ciências da computação”.

Florianópolis, 24 de outubro 2016.

Banca Examinadora:

Prof. Dr. Ricardo Azambuja Silveira
Orientador

Msc. Thiago Ângelo Gelaim
Coorientador

Prof. Dr. Elder Rizzon Santos

Prof.^a Dr.^a Jerusa Marchi

AGRADECIMENTOS

À minha família, pelo apoio durante toda a graduação. À minha mãe Angela, pelo amor incondicional e seus conselhos. Aos meus irmãos Diogo, Mariana e Maieli pelo suporte e carinho.

Ao meu coorientador Thiago Ângelo Gelaim, pela paciência e pelo tempo dedicado em me ajudar no desenvolvimento desse trabalho.

Ao meu orientador Ricardo Pereira Azambuja, por ter aceito trabalhar comigo e pelas suas dicas.

Aos membros da banca Jerusa Marchi e Elder Rizzon Santos, pelas correções e por ter aceito participar desse trabalho.

Aos meus amigos e colegas de graduação que me ajudaram durante todos esses anos. Aos colegas do laboratório de Telemedicina de Santa Catarina pelas sugestões na apresentação do TCC.

RESUMO

A utilização de agentes tem se tornado realidade em diversas áreas, porém nem sempre um único agente consegue realizar determinadas tarefas. Outro ponto importante no uso de um único agente é que atingir certos objetivos pode gerar um grande custo computacional, onde recursos não são usados de maneira eficiente. Para tratar tais problemas, o uso de vários agentes torna-se essencial, surgindo a necessidade de desenvolver um sistema multiagente, que tem como um dos principais propósitos atingir os objetivos gerais de um sistema de forma eficiente. Ao inserir diversos agentes em um ambiente, é provável que exista uma disputa pelos recursos limitados ou certas decisões que exigem que os agentes entrem em acordo sobre a próxima ação a ser executada. Para a resolução de conflitos é preciso que técnicas de negociação sejam aplicadas. Tendo em vista esses fatores, esse trabalho propõe um protocolo de negociação em sistemas multiagente para o agendamento automático de reuniões. O desenvolvimento do protótipo do protocolo está baseado na aplicação de duas estratégias de negociação e tem como objetivo aumentar a expressividade dos agentes durante a definição de compromissos.

Palavras-chave: Agentes. Sistema Multiagente. Negociação. Protocolo. Agendamento Automático de Reuniões.

ABSTRACT

The usage of agents has become reality in many fields, although not always can a single agent perform certain tasks. Another important aspect related to the usage of a single agent is that achieving some specific objectives can cause a large computational cost, where resources are not used in efficiently. To address these problems, the usage of several agents becomes fundamental, thus raising the need to develop a multi-agent system, whose aim, among others, is to achieve the global objectives of a system in an efficient way. When several agents are inserted in an environment, there may be some dispute of limited resources or certain decisions that require the agents to come into an agreement about the next action to be executed. In order to solve these conflicts, negotiation techniques must be applied. With these factors in mind, this study proposes a negotiation protocol in a multi-agent system for automatic meeting scheduler. The protocol prototype development is based on two negotiation strategy application and aims to increase agents expressiveness during commitment definition.

Keywords: Agents. Multi-Agent Systems. Negotiation. Protocol. Automatic Meeting Schedule.

LISTA DE FIGURAS

Figura 1	Interações entre SMA, adaptado de Wooldridge (2009).	25
Figura 2	Espaço de aceitação de agentes, adaptado de Jennings et al. (2001).	36
Figura 3	Arquitetura BGP, adaptado de Cao et al. (2015).	46
Figura 4	Diagrama de interação da execução do protocolo de negociação.	62
Figura 5	Fatos e estruturas da base de crenças.	65
Figura 6	Estágios da metodologia <i>STRATUM</i> , adaptado de Rahwan et al. (2007).	81

LISTA DE TABELAS

Tabela 1	Relação de hierarquia entre professores e alunos.....	40
Tabela 2	Definição das capacidades dos agentes retirada de (RAHWAN et al., 2007).....	51
Tabela 3	Adaptação do modelo de definição de estratégias/táticas do <i>STRATUM</i> para o <i>feedback</i>	59
Tabela 4	Adaptação do modelo de definição de estratégias/táticas do <i>STRATUM</i> para o esforço	61
Tabela 5	Agenda de e1.....	70
Tabela 6	Agenda de r1.....	71
Tabela 7	Agenda de r2.....	71
Tabela 8	Resultados do cenário 1.....	72
Tabela 9	Agenda de e1.....	73
Tabela 10	Agenda de r1.....	73
Tabela 11	Agenda de r2.....	73
Tabela 12	Resultados do cenário 2.....	74
Tabela 13	Relação hierárquica adaptada de Hossain e Shakshuki (2013).....	75
Tabela 14	Agenda do professor Darcy.....	75
Tabela 15	Agenda dos estudantes.....	75
Tabela 16	Adaptação do modelo de definição de estratégias/táticas do <i>STRATUM</i>	83

LISTA DE ABREVIATURAS E SIGLAS

BDI	Belief-Desire-Intention	24
KSE	<i>Knowledge Sharing Effort</i>	27
KIF	<i>Knowledge Interchange Format</i>	27
FIPA	<i>Foundation for Intelligent Physical Agents</i>	29
IA	Inteligência Artificial	38
SAR	Servidor para Agendamento de Reuniões	41
KQML	Knowledge Query Manipulation Language	42
AAR	Agendamento Automático de Reuniões	50
AGM	Alchourrón Gärdenfors e Makinson	53

SUMÁRIO

1 INTRODUÇÃO	19
1.1 MOTIVAÇÃO E JUSTIFICATIVA	20
1.2 OBJETIVO GERAL	20
1.3 OBJETIVOS ESPECÍFICOS	20
1.4 ORGANIZAÇÃO DO TEXTO	21
2 FUNDAMENTAÇÃO TEÓRICA	23
2.1 AGENTES	23
2.1.1 Sistemas multiagente	23
2.1.2 Agentes BDI	24
2.2 INTERAÇÕES EM SMA	25
2.3 COOPERAÇÃO, COLABORAÇÃO E COORDENAÇÃO EM SMA	26
2.4 COMUNICAÇÃO	26
2.4.1 Linguagens de comunicação de agentes	27
2.4.1.1 <i>Knowledge Interchange Format</i> (KIF)	27
2.4.1.2 <i>Knowledge Query and Manipulation Language</i> (KQML)	27
2.4.1.3 Linguagem de comunicação de agentes FIPA	29
2.5 NEGOCIAÇÃO	30
2.5.1 Aplicação de teoria dos jogos em SMA	31
2.5.2 Limitações de técnicas de teoria dos jogos em SMA	32
2.5.3 Leilões	33
2.5.3.1 Aplicação de leilões em SMA	34
2.5.4 Áreas de aceitação durante a etapa de negociação	35
2.5.5 Negociação baseada em argumentação	37
2.5.6 Uso de heurística em negociação entre agentes	38
2.6 ESTADO DA ARTE	38
2.6.1 <i>A Negotiation Protocol for Meeting Scheduling Agent</i>	38
2.6.2 A distributed multi-agent meeting scheduler	40
2.6.3 Towards host-to-host meeting scheduling negotiation	42
2.6.4 Automated negotiation for e-commerce decision making: A goal deliberated agent architecture for multi-strategy selection	44
3 DESENVOLVIMENTO	49
3.1 PROPOSTA DE UM PROTOCOLO DE NEGOCIAÇÃO EM SISTEMAS MULTIAGENTE	49
3.1.1 Especificação dos objetivos do agente	50
3.1.2 Descrição das capacidades do agente	50

3.1.3 Base de crenças	51
3.1.4 Definindo prioridade de assuntos e pessoas	52
3.1.5 Resposta através do <i>feedback</i>	53
3.1.6 Definição de esforço para alocação de um compromisso	54
3.1.7 Construção do ambiente	55
3.2 ESTRATÉGIAS E FUNCIONAMENTO DO PROTOCOLO	56
3.2.1 Propondo compromissos	56
3.2.1.1 Definindo a resposta	57
3.2.2 Analisando as respostas	59
3.2.3 Lidando com impasses e conjunto de horários	59
4 IMPLEMENTAÇÃO DO PROTÓTIPO DO PROTOCOLO	63
4.0.1 Interface C++/Prolog	63
4.0.2 Organização da implementação do protótipo do protocolo	64
4.0.2.1 Desenvolvimento da base de crenças	64
4.0.2.2 Acessos à base de crenças utilizando C++	65
4.0.3 Definição da estratégia utilizada	67
4.0.4 Tipos de mensagens presentes no protocolo	67
4.1 CENÁRIOS E TESTES REALIZADOS	69
4.1.1 Cenário 1	70
4.1.2 Cenário 2	72
4.1.3 Cenário 3	74
5 CONCLUSÃO	77
5.1 TRABALHOS FUTUROS	78
REFERÊNCIAS	79
6 ANEXOS	81
6.1 ANEXO A - <i>STRATUM</i> : UMA METODOLOGIA PARA DESENVOLVIMENTO DE ESTRATÉGIAS DE NEGOCIAÇÃO PARA AGENTES	81
6.1.1 Estágio I: Especificação dos objetivos do agente	82
6.1.2 Estágio II: Descrição das capacidades do agente	82
6.1.3 Estágio III: Construção do ambiente em que o agente está inserido	82
6.1.4 Estágio IV: Construção de estratégias e táticas	83
6.1.5 Estágio V: Teste e refinamento	84
6.1.6 Considerações finais sobre a metodologia	84
7 ANEXO B - ARTIGO DO TCC	85

1 INTRODUÇÃO

No estudo de agentes um dos principais objetivos é a criação de um sistema que seja autônomo, de forma que quando estiver em um ambiente dinâmico, ele consiga interagir com outras entidades e assim atingir os objetivos do sistema (WOOLDRIDGE, 2009). Atingir um certo objetivo não é o único fator importante. Assim como em outras áreas da computação, o tempo necessário e o poder computacional para finalizar uma tarefa também estão inerentes ao desenvolvimento de agentes.

Assim como no mundo real, nem todas as tarefas podem ser realizadas individualmente e mesmo que isso seja possível, demandaria um tempo considerável e um esforço de magnitude semelhante. Frequentemente requisita-se ajuda de terceiros que necessitam ter alguma espécie de conhecimento para fornecer algum auxílio (WOOLDRIDGE, 2009).

Problemas que envolvem colaboração não são exclusivos de sistemas multiagente. Pessoas interagem e colaboram entre si para a resolução de seus objetivos. Para realizar suas atividades diversos animais utilizam alguma forma de comunicação e cooperação, algumas espécies chegam a formar comunidades para garantir sua sobrevivência (KRAUS, 1997).

Tendo em vista o que foi abordado anteriormente, fica explícita a necessidade do uso de vários agentes. A área que engloba esses aspectos é conhecida como sistemas multiagente, sendo esse um dos tópicos de estudo desse trabalho. Outro ponto importante é a maneira como agentes realizam essas tarefas, quais ações cada um irá executar, como elas serão feitas e em qual ordem, ou seja, como serão definidas as ações de cada agente, de forma a tratar a maneira como eles interagem em sociedade (WOOLDRIDGE, 2009).

A existência de recursos limitados em um ambiente leva a ocorrência de disputas e conflitos entre agentes, tornando essencial a existência de algum mecanismo para a resolução desses problemas (JENNINGS et al., 2001). Para contornar conflitos e disputas é necessário empregar técnicas de negociação, que determinam a distribuição e uso de recursos em um ambiente (JENNINGS et al., 2001). Uma das propostas desse trabalho é o desenvolvimento de um protocolo de negociação para sistemas multiagente, onde cada agente tem como objetivo agendar reuniões em nome do usuário representado por ele.

1.1 MOTIVAÇÃO E JUSTIFICATIVA

É comum em um sistema multiagente existirem disputas por um mesmo recurso (JENNINGS et al., 2001). Esse tipo de impasse também está presente no cotidiano e muitas vezes pessoas utilizam algum mecanismo para chegarem em um acordo. A etapa de negociação tem se mostrado um mecanismo viável para resolução de conflitos, pelo fato de utilizar tópicos de outras áreas para definir uma estratégia (KRAUS, 1997).

O uso de agentes em tarefas executadas por humanos tem ganhado um espaço considerável nos últimos anos. A venda de itens na internet, onde agentes interagem com outros agentes ou com pessoas para vender itens em nome dos usuários é uma atividade cada vez mais comum (WURMAN; WELLMAN; WALSH, 1998).

Levando em consideração os aspectos abordados anteriormente, pretende-se desenvolver um protocolo de negociação em sistemas multiagente. Esse protocolo está voltado para o agendamento de reuniões automático, de forma que agentes troquem informações para agendar um horário que seja comum a maior parte dos envolvidos.

1.2 OBJETIVO GERAL

O objetivo do trabalho é compreender o processo de negociação em sistemas multiagente e apresentar, como estudo de caso, uma proposta de um protocolo de negociação aplicado a um sistema de controle para agendamento de reuniões baseado em agentes.

1.3 OBJETIVOS ESPECÍFICOS

Os seguintes objetivos específicos são definidos para atingir o objetivo principal:

- Analisar os principais aspectos de comunicação relacionados às interações entre agentes;
- Identificar e compreender as principais técnicas atualmente utilizadas nos processos de negociação em sistemas multiagente;
- Implementar, como estudo de caso, um protótipo de um protocolo de negociação entre agentes para uma aplicação de agendamento automático de reuniões;

- Definir um conjunto de cenários para verificar a expressividade e as principais limitações do protocolo desenvolvido.

1.4 ORGANIZAÇÃO DO TEXTO

Esse trabalho está dividido em fundamentação teórica, estado da arte, desenvolvimento, resultados obtidos e conclusão. A fundamentação teórica tem o propósito de mostrar os principais aspectos de negociação, coordenação e cooperação nas interações entre os agentes. As principais pesquisas que envolvem o agendamento automático de reuniões estão na seção sobre estado da arte. No desenvolvimento apresenta-se as características do agendamento automático de reunião, proposta e funcionamento do protocolo de negociação, cenários utilizados para testes e resultados obtidos. Na conclusão aborda-se os principais pontos utilizados para o desenvolvimento do protocolo, bem como o que os resultados representam. Os aspectos que foram desconSIDERADOS no protocolo e que podem originar trabalhos futuros também encontram-se na conclusão.

2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo serão apresentados os principais tópicos que envolvem o desenvolvimento desse trabalho. A seção 2.1 e suas subseções apresentam definições sobre agentes e aspectos relacionados a sua arquitetura. As interações em sistemas multiagente são tratadas na seção 2.2. A seção 2.4 visa abordar aspectos relacionados a comunicação entre agentes. Por fim, a seção 2.5 trata as principais características de negociação, levando em consideração como os conflitos são resolvidos em sistemas multiagente.

2.1 AGENTES

A definição do que é um agente é algo que está em constante discussão por estudiosos da área, porém é possível estabelecer diversas características essenciais no desenvolvimento de agentes. De acordo com Wooldridge, Jennings et al. (1995), Wooldridge (2009), as seguintes propriedades são necessárias:

- autonomia: agentes operam sem intervenção direta de humanos ou algum outro artifício e tem certo controle sobre suas ações e estado interno;
- habilidades sociais: interagem entre si e possivelmente com humanos. A comunicação depende de algum tipo de linguagem;
- reatividade: agentes interagem com o ambiente (mundo real, interface gráfica, grupo de agentes, e possivelmente uma combinação de todos) e respondem a esses estímulos levando um certo tempo;
- proatividade: agem de forma orientada a objetivos, ou seja, não respondem apenas a estímulos do ambiente, levam em consideração a possibilidade da ocorrência de um determinado cenário e agem de forma antecipada.

2.1.1 Sistemas multiagente

Um sistema multiagente consiste em um conjunto de agentes que interagem entre si, através da troca de mensagens, por algum proto-

colo de comunicação de redes de computadores. Tipicamente, agentes representam pessoas ou entidades que possuem diferentes objetivos e motivações (WOOLDRIDGE, 2009). Muitas vezes as interações entre agentes envolvem disputas por um mesmo recursos, sendo necessário empregar técnicas de negociação para chegar-se a um consenso (JENNINGS et al., 2001).

2.1.2 Agentes BDI

A arquitetura *Belief-Desire-Intention* (BDI) foi proposta por Bratman (1987) como um modelo filosófico sobre a teoria do raciocínio prático. Que tem como objetivo modelar atitudes mentais, sendo elas: crenças, desejos e intenções. Segundo Wooldridge (2009), o raciocínio prático é uma maneira de decidir quais ações devem ser tomadas, levando em consideração os desejos, valores e crenças de um agente. A maneira como o raciocínio prático é usado ocorre na maioria das vezes em duas etapas, a deliberação e o raciocínio meio-fim. A deliberação acontece quando imagina-se em qual estado do mundo deseja-se chegar, já o raciocínio meio-fim tem como objetivo definir como atingir esse determinado estado. Um bom exemplo para ilustrar tais conceitos é aquele onde se quer definir qual será o destino de uma viagem, após essa etapa são estipulados alguns aspectos relevantes, sendo eles: qual tipo de transporte, qual lugar será mais adequado para hospedar-se, os preços, a quantidade de dias e etc (WOOLDRIDGE, 2009).

Bratman (1987) e Wooldridge (2000) mostram que as intenções são atitudes mentais que tem um papel relevante no raciocínio prático:

- Intenções direcionam raciocínio meio-fim: dada uma intenção, o agente tentará realizá-la de alguma forma;
- Intenções persistem: existem três motivos para que um agente deixe de querer realizar uma intenção: o agente acredita que ela foi realizada, a intenção não pode ser realizada ou o motivo que originou a intenção não existe mais;
- Intenções restringem deliberações futuras: o agente não possui intenções que são inconsistentes entre si;
- Intenções influenciam as crenças sobre as quais os raciocínios práticos são baseados: dada a existência de uma intenção, planos futuros podem considerar que ela será realizada.

2.2 INTERAÇÕES EM SMA

Segundo Wooldridge (2009), agentes encontram-se em diferentes esferas de influência, ou seja, diferentes agentes irão agir em diferentes partes de um ambiente e consequentemente serão influenciados pelos estímulos de tal. Em diversos casos eles devem formar associações para resolver um determinado problema, essas associações são conhecidas como alianças (KRAUS, 1997).

Em casos onde ocorre o compartilhamento da mesma esfera de influência, algum relacionamento de dependência entre esses agentes pode surgir. Esse relacionamento é muitas vezes definido pelo tipo de interação existente entre agentes (WOOLDRIDGE, 2009).

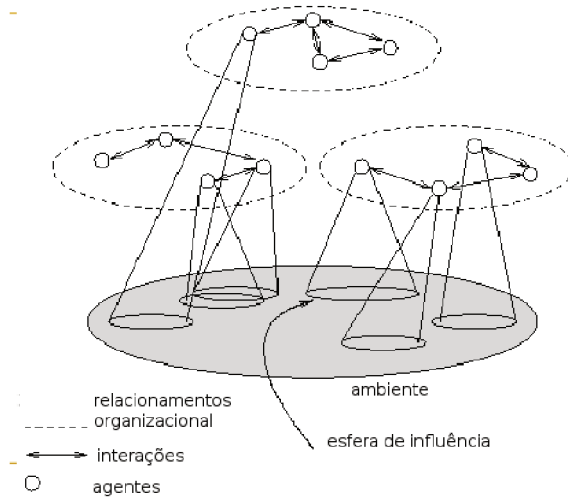


Figura 1 Interações entre SMA, adaptado de Wooldridge (2009).

Na figura 1 é possível notar que agentes que compartilham uma mesma esfera de influência e interagem entre si podem necessitar de um mesmo recurso do ambiente, tal ocorrência poderá acarretar em disputas e conflitos entre agentes, tornando-se necessário o uso de algum mecanismo de negociação para chegar em um acordo (WOOLDRIDGE, 2009).

2.3 COOPERAÇÃO, COLABORAÇÃO E COORDENAÇÃO EM SMA

O uso de sistemas multiagente em diversas aplicações tem se tornado notável, tal fato faz com que a implementação de agentes também siga esse fluxo. Inevitavelmente um agente pode interagir com outro agente que não siga os mesmos protocolos que ele ou que não compartilhe das mesmas crenças e objetivos (WOOLDRIDGE, 2009). Para garantir que agentes ajam de forma autônoma, eles devem organizar-se dinamicamente para atingirem um estado de mundo desejado. Para chegarem em tal estado mecanismos de coordenação, colaboração e cooperação podem ser empregados, assim como ocorre no cotidiano (KRAUS, 1997).

Wooldridge (2009) define que um problema de coordenação consiste em gerenciar as interdependências entre as atividades dos agentes. Segundo Zhang et al. (2009), as atividades dos agentes são dirigidas pelos seus objetivos. Essas propriedades definem se um agente têm um comportamento competitivo ou cooperativo.

(ZHANG et al., 2009) diferenciam colaboração e cooperação em SMA da seguinte forma:

- Um SMA que contém agentes com objetivos distintos ou objetivos individuais, geralmente, usa colaboração para chegar em um determinado estado de mundo;
- Quando agentes possuem objetivos comuns, então pode-se definir o SMA como cooperativo.

Apesar de todos os agentes terem o objetivo de alocar reuniões para seus usuários, existem fatores que evidenciam que nem sempre todos os agentes estão de acordo. Se um usuário prefere ter suas reuniões alocadas para a parte da manhã, o agente que o representa irá levar em consideração tal preferência, desconsiderando a decisão de outros agentes. Mesmo priorizando o usuário, agentes devem seguir um protocolo e colaborar entre si para que seja possível o agendamento de reuniões.

2.4 COMUNICAÇÃO

Segundo Wooldridge (2009), agentes interagem com outros agentes tendo um dos principais propósitos alterar o estado mental dos envolvidos. O desenvolvimento da comunicação em SMA está fundamentado na teoria atos de fala, onde a comunicação é tratada como uma ação realizada pelo agente.

De acordo com Austin (1975), certas palavras têm características que remetem à ações, ou seja, alteram o estado do mundo de forma semelhante à ações físicas. Alguns verbos podem ser classificados como verbos performativos. Os principais são verbos que remetem para as seguintes características: pedir, informar e prometer (AUSTIN, 1975).

2.4.1 Linguagens de comunicação de agentes

Com o propósito de garantir a troca de informação e conhecimento entre agentes, diversas linguagens começaram a ser desenvolvida após 1990. Inicialmente, uma das iniciativas que teve maior destaque foi a *Knowledge Sharing Effort* (KSE) . Seu desenvolvimento estava focado na troca de conhecimento entre sistemas inteligentes autônomos (WOOLDRIDGE, 2009). Através da *KSE* surgiram duas linguagens de comunicação, sendo elas o *Knowledge Interchange Format* (KIF) e *Knowledge Query and Manipulation Language* (KQML).

2.4.1.1 *Knowledge Interchange Format* (KIF)

Segundo Labrou, Finin e Peng (1999), a KIF foi criada tendo como um dos principais objetivos a representação de objetos relacionados a banco de dados, agentes inteligentes e outros sistemas computacionais, ou seja, uma maneira de representar propriedades de um domínio específico. Outro fator muito importante é que a KIF serviria como uma linguagem intermediária entre outras linguagens de representação de conhecimento (LABROU; FININ; PENG, 1999; WOOLDRIDGE, 2009).

Segundo Wooldridge (2009), a KIF está fortemente baseada em lógica de primeira ordem e procura expressar o conteúdo de uma mensagem, deixando de lado aspectos relacionados à troca de mensagens. Segue abaixo um exemplo adaptado de Wooldridge (2009) de uma mensagem KIF que representa uma afirmação da temperatura de m1:

(= (temperatura m1) (escalar 83 celsius))

2.4.1.2 *Knowledge Query and Manipulation Language* (KQML)

Para Labrou, Finin e Peng (1999), o KQML é uma linguagem de comunicação orientada a mensagens. Também é um protocolo para troca de informação independente de diversos aspectos. Sendo eles:

mecanismos de transporte (TCP/IP, SMTP, IIOP e outros), conteúdo da linguagem (KIF, SQL, Prolog e outros) e independente da ontologia do conteúdo da mensagem. Wooldridge (2009) define que um dos principais propósitos da KQML é definir um padrão no formato das mensagens trocadas entre agentes e não preocupa-se com o conteúdo da mensagem.

Segue abaixo um exemplo de uma mensagem KQML adaptada de Wooldridge (2009), onde é solicitado os preços das ações da IBM:

```
(ask-one
  :content (preço IBM ?preço)
  :receiver servidor-de-mercado-de-ações
  :language LPROLOG
  :ontology NYSE-TICKS
)
```

Segundo (LABROU; FININ; PENG, 1999), pode ser identificadas três camadas organizacionais do KQML, sendo elas as seguintes:

- Conteúdo: possui de fato o conteúdo de uma mensagem, onde uma representação em qualquer linguagem pode ser inserida.
- Comunicação: descreve os parâmetros de comunicação de baixo nível. Possui informações relacionadas ao emissor da mensagem;
- Mensagem: essa camada é responsável por codificar o que pretende-se transmitir. Ela determina o tipo de interação que agentes, que fazem o uso do KQML, irão realizar. Por meio dessa camada é possível identificar qual ato de fala o emissor irá utilizar.

Segundo Wooldridge (2009), a KQML foi muito utilizada por muitos anos na comunidade de desenvolvimento de SMA. Entretanto, existem alguns pontos negativos nela, sendo eles os seguintes:

- A semântica da KQML nunca foi definida de forma rigorosa, o que impossibilitava de estipular se um agente estava realmente utilizando a linguagem;
- Mecanismos de transportes também não foram precisamente definidos;
- O conjunto básico de performativos da KQML dificultava a interoperabilidade;

2.4.1.3 Linguagem de comunicação de agentes FIPA

Para tentar resolver todos os problemas levantados pela comunidade, começou-se a desenvolver uma nova linguagem pela FIPA. Criado em 1995 pela *Foundation for Intelligent Physical Agents* (FIPA). O principal propósito da fundação é criar padrões para desenvolvimento agentes, de forma a atingir alto grau de interoperabilidade (WOOLDRIDGE, 2009).

Segundo Labrou, Finin e Peng (1999), a linguagem de comunicação de agentes desenvolvida pela FIPA é muito similar a KQML. Tendo a sintaxe idêntica a KQML, variando apenas algumas primitivas de comunicação. A linguagem consiste de um conjunto de tipos de mensagens, os efeitos nas atitudes mentais tanto agentes emissor quando do receptor. A especificação da linguagem descreve cada ato de comunicação de forma narrativa e semântica, sendo essa última forma baseada em lógica modal.

Para tratar os problemas de semântica do KQML, os desenvolvedores definiram a semântica da linguagem baseada em uma outra linguagem, a SL. Essa linguagem permite a representação de crenças, desejos, crenças incertas e as ações realizadas pelo agente (WOOLDRIDGE, 2009). Abaixo encontra-se um exemplo adaptado de Wooldridge (2009) de uma mensagem da linguagem de comunicação de agentes:

```
(inform
  :sender agente1
  :receiver agente2
  :content (preço item2 150)
  :language sl
  :ontology hpl-auction
)
```

É notável a evolução que as linguagens de comunicação sofreram durante os últimos anos. A intersecção entre a teoria e prática também é um ponto que ficou explícito durante essa seção. Na próxima seção será mostrado os principais mecanismos de negociação para SMA.

A importância da comunicação em negociação é um fator relevante para que o rumo de um acordo seja definido, já que é por meio da comunicação que o agente consegue convencer outros agentes ou pessoas. Também é válido ressaltar que uma boa estratégia de negociação apoia-se em uma boa representação de informação, que como visto nessa seção tem grande utilidade para o desenvolvimento de SMA.

2.5 NEGOCIAÇÃO

Segundo Jennings et al. (2001), a negociação pode ser considerada a interação mais importante entre agentes, já que é um mecanismo para gerenciar interdependência em tempo de execução entre agentes. Jennings et al. (2001) também definem a negociação como o processo onde um grupo de agentes chegam em acordo sobre algo. Suas interações consistem basicamente em realizar propostas, apresentar opções para um problema, conceder recursos e por fim chegar em um acordo sobre o que deve ser feito (WOOLDRIDGE, 2009).

Kraus (1997) define as propriedades que devem ser postas em análise ao escolher qual técnica de negociação deve ser aplicada:

- O nível de cooperação entre agentes: grupo de agentes que trabalham de forma cooperativa para alcançar um mesmo objetivo ou agentes que possuem vontades próprias e tentam maximizar seus próprios ganhos. Existem casos intermediários em que agentes não benevolentes trabalham de forma cooperativa para atingir um objetivo global;
- Regulamento e protocolo: ambientes onde todos os agentes seguem um mesmo regulamento e protocolo ou situações onde nenhum deles foi pré-definido;
- Quantidade de agentes: disponibilidade de quantidade elevada de agentes ou poucos agentes que utilizam mecanismos de coordenação e cooperação e tentam maximizar o uso de recursos disponíveis;
- Tipo de agentes: sistemas compostos apenas por agentes que interagem entre si ou ambientes onde existem interação com humanos e outras máquinas;
- Comunicação e custo de computação: disponibilidade de recursos e o quanto é gasto em termos computacionais para realizar comunicação.

De acordo com Kraus (1997) nem sempre é possível existir um cenário onde as propriedades estabelecidas anteriormente estão totalmente separadas. Kraus (1997) utiliza um exemplo em que agentes tentam realizar a venda de um produto para outros agentes ou para pessoas interessadas. As interações entre os agentes podem ser divididas em:

- Um agente tentará vender algo para outro agente, ou seja, ambos serão beneficiados por tal transação. O agente vendedor tentará maximizar suas vendas e o comprador tentará obter um produto que a sua empresa ainda não possui. Existe um objetivo distinto entre cada agente, porém ambos beneficiam-se caso usem alguma espécie de cooperação;
- O agente vendedor tentará vender algo para alguma pessoa, nesse caso uma pessoa iria preferir uma negociação não muito estruturada, ou seja, o não envolvimento de protocolos pareceria mais natural nesse contexto;
- Dois vendedores trabalham juntos para maximizar o lucro da empresa. Nesse caso existe um objetivo global entre agentes.

Através do cenário acima conclui-se que em diversos momentos existe uma intersecção entre as propriedades e que o desenvolvimento de um protocolo encarregado de cobrir o cenário acima deve fazer uso de algumas características que são tratadas por Kraus (1997): Um protocolo de negociação deve garantir sucesso, onde eventualmente um acordo deverá ser feito pelos agentes envolvidos. A saída do protocolo deve maximizar os ganhos de todos os agentes participantes e também deve ser levado em consideração que não exista uma saída em que um agente consiga um resultado melhor do que já foi proposto, ou seja, um dos agentes leve vantagem por meio da desvantagem de outros agentes. Um mecanismo de negociação deve ser intuitivo, fazendo com que um agente consiga determinar, de forma fácil, uma estratégia ideal. A comunicação entre agentes deve ser tolerante à falhas, caso alguma mensagem seja perdida deve existir algum mecanismo encarregado de notificar tal falha aos envolvidos.

Pode ser empregado o uso de teoria dos jogos para tratar casos em que seja necessário utilizar algum mecanismo de negociação. A próxima subseção apresenta de maneira simplificada a aplicação de teoria dos jogos em SMA e as suas consequências.

2.5.1 Aplicação de teoria dos jogos em SMA

Quando existem problemas onde agentes possuem interesses próprios e querem sempre maximizar seus ganhos, é recomendado a utilização de técnicas utilizadas em teoria dos jogos (KRAUS, 1997). Sua expressividade está no uso de modelos matemáticos para representar situações de conflitos entre indivíduos. Os modelos usados em teoria

dos jogos são representações abstratas de situações reais que representam os indivíduos que possuem visões e objetivos diferentes. Em casos onde agentes precisam dividir recursos e realizar tarefas em grupos, e também não possuem um objetivo global, são usadas técnicas de negociações para resolver os conflitos (MIHAYLOV, 2012; KRAUS, 1997).

Mihaylov (2012) sugere um protocolo em que cada agente faz uma oferta sobre um determinado recurso e a cada iteração uma votação é feita. O tempo decorrido nessa negociação também seria um fator decisivo na implementação do protocolo. A formalização do protocolo pode ser descrita da seguinte forma: cada agente realiza uma ação de oferta em um instante t , sendo $t \in T$. Se todos os agentes aceitam a proposta, então a negociação termina. Caso um agente aceite e outro rejeite, então uma nova iteração de negociação é feita.

Apesar de existir todo um mecanismo que possibilite essa atividade de negociação, Mihaylov (2012) aponta alguns problemas que devem ser levados em consideração:

- Adaptar essa técnica para o domínio do problema, levando em consideração aspectos fundamentais presentes em sua resolução;
- Desenvolver estratégias de equilíbrio;
- Usar técnicas que sejam viáveis em relação ao tempo e ao processamento necessário;
- Obter funções de utilidades para o problema proposto.

2.5.2 Limitações de técnicas de teoria dos jogos em SMA

Apesar de teoria dos jogos estar diretamente ligada às interações e decisões em grupo em sistemas multiagente, existem algumas limitações impostas pelo seu modo de funcionamento. Tornando inviável sua aplicação no desenvolvimento de protocolos de negociação e em vários cenários (SIERRA et al., 1997).

Em teoria dos jogos uma estratégia é definida como uma possibilidade selecionada a partir de todas as possíveis estratégias de um problema. A sua aplicação em cenários reais acarreta em um crescimento exponencial das possibilidades de estratégias que um agente pode realizar (JENNINGS et al., 2001).

Essa quantidade elevada de possibilidade acaba gerando um problema entre a teoria e a prática, já que a busca pela melhor estratégia pode demorar para ser definida. Segundo Jennings et al. (2001), a busca

pela melhor estratégia em um grande conjunto de possibilidades levaria um agente à raciocinar por um tempo indeterminando, tornando-se inviável sua aplicação.

A teoria dos jogos assume que os agentes conhecem todas as possibilidades de um cenário, assim como o resultado de cada ação aplicada ao cenário. Porém, em diversos casos isso não é verdade, o que torna seu uso pouco abrangente (JENNINGS et al., 2001).

Para contornar esses problemas, originados pelo uso de técnicas de teoria dos jogos, é preciso utilizar mecanismos que não sejam baseados na exploração das possibilidades em diferentes iterações da negociação.

2.5.3 Leilões

Um protocolo de negociação muito usado para resolução de conflitos são os leilões, que servem para definir como agentes irão utilizar recursos que são compartilhados e limitados. Por mais simples que a proposta de um leilão possa parecer, existem diversos tipos de leilões que são implementados para que agentes cheguem em um acordo (GUTTMAN; MOUKAS; MAES, 1998). Essas diferentes propostas de leilões atuam sobre uma mesma “base”, ou seja, existe um indivíduo oferecendo um item, no caso de agentes estamos lidando com recursos, e apostadores que desejam obtê-los (WURMAN; WELLMAN; WALSH, 1998).

Assim como em leilões tradicionais onde de um lado há um vendedor querendo maximizar o ganho de um item e do outro um comprador interessado em minimizar o preço que irá pagar pelo item. Agentes irão utilizar alguma estratégia, que leve em consideração as regras do protocolo usada na negociação, para otimizar seus resultados (WOOLDRIDGE, 2009).

Para Guttman, Moukas e Maes (1998), negociações podem ser descritas como um processo para resolver conflitos entre dois ou mais interessados em um objetivo único e mutualmente exclusivo, no caso de agentes pode ocorrer a associação desse objetivo a um recurso limitado no ambiente. Wooldridge (2009) defende que a estratégia não é o único fator que deve ser utilizado na hora de implementar protocolos de leilões, sendo essas propriedades citadas abaixo:

- Valor privado: todo agente define um valor interno para uma determinada tarefa, ou seja, o agente tem alguma razão, que não é disponibilizada para outros agentes, para definir o quanto ele deve se esforçar para conseguir um recurso ou tarefa;

- Valor correlato: o agente que deseja obter um recurso leva em consideração o quanto outros agentes também querem tal recurso.

Wurman, Wellman e Walsh (1998) apresenta diversos tipos de leilões, sendo o mais comum aquele em que o ganhador é o que realizou o maior lance, conhecido como leilão inglês¹. Outra tipo de venda de itens é o conhecida como leilão holandês, onde o preço inicial de um item começa com um valor elevado e então esse valor vai sofrendo decréscimos até alguém aceitar o valor proposto. É válido lembrar que em ambos os mecanismos apresentados acima, todos os participantes conseguem saber quais os lances estão sendo propostos. Já em leilão fechado todos os lances são realizados de forma privada. E em leilão de Vickrey o interessado que realizar o maior lance irá pagar o valor proposto pelo segundo maior lance.

Wooldridge (2009) apresenta um cenário em que agentes não tem conhecimento dos lances realizados por outros agentes e caso um agente realize uma oferta muito maior que seu real valor, a propriedade que garante o equilíbrio em negociação será mantida, já que o agente que realizou a segunda maior aposta está mais próximo do valor real e não “extrapolou” o valor real.

Todas os mecanismos de leilões apresentados partem da premissa que participantes são exclusivamente compradores ou vendedores. De acordo com Guttman, Moukas e Maes (1998), existem leilões onde esses papéis não são exclusivos, sendo esse um mecanismo muito usado para venda e compras de ações, onde lances são zerados conforme o decorrer do tempo ou ao fim de rodadas.

2.5.3.1 Aplicação de leilões em SMA

Guttman, Moukas e Maes (1998) abordam alguns exemplos de aplicações que usam técnicas de negociação em leilões que são realizados na internet. O primeiro deles é um sistema multiagente web chamado Kasbah, onde usuários criam agentes que auxiliam na venda e compra de itens. Um usuário que deseja realizar alguma dessas duas tarefas cria um agente e define o seu tipo de estratégia e então o adiciona em uma espécie de mercado público. Após essa etapa, o agente irá iniciar a busca por outros agentes que estejam interessados em alguma oferta ou que estejam ofertando algo. Ao encontrar algum agente interessado em vender ou comprar algo, ocorre início ao processo de negociação.

¹english open outcry

O objetivo de cada agente é obter um acordo aceitável, conforme definido pelo usuário, e então notificar ao usuário sobre as seguintes opções: maior oferta, menor oferta e data estipulada para completar a transação. O Kasbah implementa um protocolo estrito e direto, após dois agentes iniciarem a negociação, o agente que desejar comprar um item oferece um valor e o vendedor apenas responde se aceita ou não a oferta. Outro ponto interessante desse sistema é que existem apenas três estratégias disponíveis e elas correspondem ao valor da função que representa o preço do produto com o decorrer do tempo, sendo elas: linear, quadrática ou exponencial. Um dos motivos por manter a simplicidade do sistema é a fácil aceitação pelos usuários, ele tem sido bastante usado no instituto de tecnologia de Massachusetts para venda e compra de livros, álbuns de música e etc.

Outro exemplo abordado rapidamente por Guttman, Moukas e Maes (1998), é o *AuctionBot* que se assemelha muito com o Kasbah, onde sua principal diferença é que os leilões ocorrem de forma multilateral, ou seja, vários agentes podem realizar ofertas em um item.

2.5.4 Áreas de aceitação durante a etapa de negociação

De acordo com Jennings et al. (2001), a negociação pode ser vista como a procura distribuída por um acordo em um espaço. Conforme mostra a figura 2, cada agente possui uma área de aceitação. É importante lembrar que assim como humanos, agentes também podem tornar-se mais rigorosos ou flexíveis, ou seja, sua área de aceitação sobre algo pode diminuir ou expandir, assim como sua função de utilidade. Agentes também definem pontos para potenciais acordos e esses pontos movem-se dentro do espaço de aceitação de cada agente e assim como as mudanças dos espaços, esses pontos seguem a mesma regra durante a negociação. O fim da etapa de negociação ocorre quando um número mínimo de acordos é definido ou quando o protocolo dita o fim desse processo (JENNINGS et al., 2001).

Levando em consideração a figura 2 e o que foi mostrado anteriormente, fica claro que o agente A1 e o agente A2 não irão chegar em um acordo, já que não existe nenhum ponto de potenciais acordos entre eles. De fato, A2 não possui nenhum ponto de acordo que represente uma oferta e nenhuma das ofertas de A1 está na área de interesse de A2. Já na negociação entre A1 e A3 existe um ponto de interesse que está sendo compartilhado por ambos, tendo assim um possível acordo sobre um assunto ou sobre um conjunto de assuntos (JENNINGS et al.,

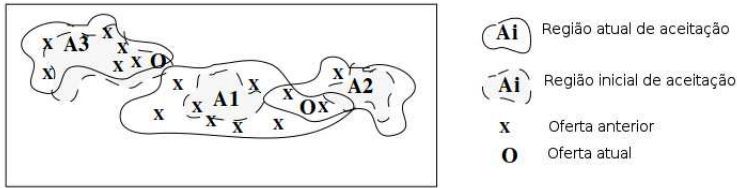


Figura 2 – Espaço de aceitação de agentes, adaptado de Jennings et al. (2001).

2001).

É visto que para realizar acordos um agente deve ser capaz de fazer propostas (definir pontos em um espaço de aceitação) e responder a propostas. Porém, apenas esses dois mecanismos são inefficientes, já que diversas propostas devem ser realizadas para se tentar um acordo. É importante ressaltar que os agentes estão inseridos em um contexto em que o tempo e recursos são limitados (JENNINGS et al., 2001; AKNINE; ARIB; BOUKREDERA, 2014).

Para Aknine, Arib e Boukredera (2014), adotar um modelo de negociação mais deliberativo é mais vantajoso, já que participantes podem, dinamicamente, estender aspectos do que está sendo proposto através da adição ou remoção de informações que possam estar atrasando a chegada a um consenso. Jennings et al. (2001) utiliza o termo *feedback* para abordar o caso citado anteriormente, com isso é possível ter uma indicação do quão próximo da área de aceitação a proposta está.

Outro mecanismo usado é a contraproposta, o diferencial de uma contraproposta é que o agente pode explicitar partes de suas intenções sobre a proposta, então o ofertante pode decidir se é de seu interesse continuar ou não a negociação (JENNINGS et al., 2001).

Jennings et al. (2001) defende que tanto o *feedback* quanto a contraproposta irão apenas revelar aspectos sobre o tópico que está sendo negociado, sendo assim, a resposta não deve conter tantas informações a ponto de deixar a comunicação entre os agentes com alguma sobrecarga. Por fim, Jennings et al. (2001) mostra que apesar da utilização de um leilão ser algo mais comum, existem algumas limitações nela. Os principais pontos positivos no uso de algum dos mecanismos mostrados anteriormente são:

- Justificar sua posição na negociação: um agente sempre tem um

motivo específico para aceitar ou rejeitar uma proposta, se a outra parte interessada souber seus motivos fica mais simples fazer outra tentativa ou tentar eliminar os empecilhos envolvidos na negociação para futuras interações com outros agentes;

- Persuadir mudanças: sabendo os pontos que impedem a interseção entre ponto de oferta e região de aceitação, fica mais simples do agente ofertante realizar uma tentativa de mudar a região de aceitação do outro agente. Busca-se alterar a posição de um agente através de argumentos, de modo que uma nova proposta contenha algumas das alterações sugeridas pelo outro agente. Essas estratégias são fundamentais para convencer o agente a aceitar a proposta.

Apesar de estar em um contexto que assemelha-se a um leilão, e muitas vezes de fato é um leilão, nota-se que não é suficiente em ambientes onde o tempo é estrito e recursos são limitados (WOOLDRIDGE, 2009). Os tópicos apresentados acima mostram agentes utilizando estratégias para defender sua posição perante a uma negociação.

2.5.5 Negociação baseada em argumentação

As limitações apresentadas na subseção 2.5.2 são características que podem ser ignoradas em cenários específicos, onde o espaço de possibilidades não é tão elevado. Porém, em outros casos é necessário o uso de argumentação para contornar esses problemas, o que torna possível o “encurtamento” do números de iterações da etapa de negociação.

Segundo Sierra et al. (1997), há diversas maneiras de adicionar propriedades de argumentação em propostas, sendo que as três principais são:

1. ameaça: o que irá ocorrer com o agente caso ele rejeite a proposta corrente, levantando aspectos negativos;
2. recompensa: caso a proposta seja aceita, quais serão os benefícios que o agente terá;
3. apelo: quais as principais razões para que o agente aceite a proposta desse agente;

Essas técnicas basicamente ilustram uma tentativa de um agente mudar as preferências, crenças e objetivos de um outro agente, fazendo com que ele mude sua área de aceitação em direção a proposta realizada (JENNINGS et al., 2001; AKNINE; ARIB; BOUKREDERA, 2014).

2.5.6 Uso de heurística em negociação entre agentes

Como apresentado anteriormente, o uso de técnicas de teoria dos jogos possui alguns problemas e limitação em relação à sua aplicação. Jennings et al. (2001) mostra que um dos principais motivos para adotar técnicas heurísticas é contornar o problema de crescimento exponencial no espaço de possibilidades, de forma que não ocorra a busca de uma possibilidade de forma exaustiva. A principal desvantagem de métodos heurísticos em SMA, e também em outras áreas de IA, é que a solução encontrada pode não ser ótima (JENNINGS et al., 2001).

Os modelos que usam heurística focam em suposições próximas da realidade daquele cenário. Essa propriedade faz com que seu uso seja mais expressivo, devido a sua aplicabilidade em diversos cenários (SIERRA et al., 1997).

É possível definir a heurística em negociações de SMA como um mecanismo de deliberação, onde a tomada de decisão de um agente será centrada na definição de valores para pontos em um espaço de aceitação, ou seja, o espaço de aceitação será quantificado. Esse valor será atribuído por cada agente e ele reflete sua preferência que é expressada através de sua função de utilidade (SIERRA et al., 1997). O fim da negociação ocorre quando agentes chegam em um acordo ou quando o tempo de negociação é excedido (WOOLDRIDGE, 2009).

2.6 ESTADO DA ARTE

Nessa seção serão apresentadas implementações e estratégia de negociação utilizadas em agendamento automático de reuniões.

2.6.1 *A Negotiation Protocol for Meeting Scheduling Agent*

Hossain e Shakshuki (2013) têm como propósito construir um agente que seja capaz de agendar compromissos para um usuário. Esse agente também leva em consideração as preferências do usuário para definir seus compromissos.

Uma das características do protocolo para agendamento de reuniões é a circulação de mensagens entre todos os participantes de uma determinada reunião, facilitando assim a verificação do convite de todos os membros envolvidos. As principais características do protocolo seguem abaixo:

- $A = \{A_1, A_2, A_3, \dots, A_n\}$ é o conjunto de agentes;
- A_r = Agente requisitante, A_p = agentes que irão participar da reunião, A_h = agente que irá hospedar a reunião;
- M = assunto geral da reunião ou detalhes envolvidos no compromisso;
- M_d = detalhes da reunião, M_s = assunto da reunião;
- T_s = Data e hora da reunião;
- T_d = Duração da reunião;
- D = Decisão do convite;
- D_a = Confirmação, D_r = Rejeição;
- I = Identificador da mensagem;
- I_i = Indica estado da mensagem. Por exemplo: $I_i = 1$ quando a mensagem foi enviada para outros agentes;
- I_r = resposta da reunião, I_c = contra proposta, I_{ack} = reconhecimento da mensagem;
- ID = identificador da reunião;
- R = comentários (opcional);

Quando um agente deseja enviar convite para marcar uma reunião, a mensagem terá os seguintes parâmetros: $\langle I_i, A_h, A_p, M_s, M_d, T_s, T_d, R \rangle$. Já um mensagem que contenha a resposta de um convite segue uma das três abordagens:

1. $\langle ID, I_r, A_r, D_a, R \rangle$ para aceitar uma proposta;
2. $\langle ID, I_r, A_r, D_r, R \rangle$ para rejeitar uma proposta;
3. $\langle ID, I_c, A_r, T_s, R \rangle$ propor mudanças no convite;

Conforme mencionado anteriormente, a questão de preferência de cada usuário é um fator que é levado em consideração pelo motor de inferência. O exemplo utilizado também aborda essa propriedade, de forma que uma hierarquia é definida entre professores e estudantes. A relação entre professores e estudantes pode ser vista na tabela 1:

No cenário acima, é comum que os estudantes aceitem o convite do professor Darcy, porém eles estão mais envolvidos com o professor

Tabela 1 – Relação de hierarquia entre professores e alunos.

E-mail	Cargo	Nível
elhadi.shakshuki@acadiau.ca	Professor	1
darcy.benoit@acadiau.ca	Professor associado	2
mozammal.hossain@acadiau.ca	Mestrando	3
wael.alghamdi@acadiau.ca	Mestrando	3
qianli.shu@acadiau.ca	Graduando	4

Elhadi, ou seja, ele possui maior prioridade do que o Darcy. A melhor opção para os estudantes seria esperar uma resposta do professor Elhadi e então usá-la para tomar uma decisão. O agente que representa um estudante tem duas opções sobre o comportamento a ser adotado:

1. apresentar um comportamento individual, onde sua decisão é baseada apenas nas preferências do usuário;
2. mostrar um comportamento mais colaborativo, baseando sua decisão nas preferências do grupo.

Assumindo que esses agentes tenham um comportamento colaborativo, as decisões dos outros participantes deverão ser monitoradas e a partir disso uma resposta é elaborada. Já um comportamento individual não iria necessitar dessa verificação. Para o agente tomar uma decisão ele pode precisar de duas informações:

- Informações básicas de outros participantes que serão usadas para criar um perfil informativo;
- Ter acesso a decisão de outros participantes durante o processo de negociação;

Por fim, é mostrado o resultado do convite e tanto o professor Elhadi quanto seus estudantes aceitam o convite. Isso ocorre pois todos os participantes do convite tem conhecimento das decisões, de forma que o agente que representa os alunos sabe que o professor Elhadi aceitou, sendo esse um fator decisivo para o definir o resultado do convite.

2.6.2 A distributed multi-agent meeting scheduler

Shakshuki et al. (2008) também têm seu foco voltado para criação de um agente que agende compromissos, eventos e reuniões para

um usuário. Porém, o diferencial de seu trabalho está no fato de como lidar com conflitos de horários entre compromissos. Sua estratégia está voltada para a sobrescrita da preferência do usuário, de forma que ao propor novos compromissos o agente considera horários que foram reservados pelo usuário como candidatos para o horário da reunião.

A arquitetura do sistema multiagente foi desenvolvida para facilitar a interação entre as preferências e calendário do usuário. O paradigma usado é o servidor-cliente, que é formado por um Servidor para Agendamento de Reuniões (SAR) e um agente para a alocação de reuniões.

O AAR contém seis módulos: visualizador de calendário, gerenciador, aprendizagem, agendador, gerador de e-mail e base de dados interna. O módulo relativo ao calendário é uma interface gráfica disponível ao usuário. O agendador é encarregado de negociar com outros agentes quando choques de horário surgem. Todas as decisões do usuário são escritas e monitoradas pelo módulo de aprendizagem, que utilizará essas informações para futuras decisões em uma negociação. O agendador tem como função procurar por horários disponíveis e enviar essas opções para o gerenciador. O gerador de e-mail irá enviar mensagens para os participantes de cada reunião.

O SAR possui os seguintes módulos: componente de comunicação, componente de base de conhecimento, gerador de mensagens, extrator de mensagens e base de dados de *backup*. O componente de comunicação tem como função estabelecer a conexão entre agentes, bem como a troca de mensagens e sua interpretação. A base de dados de *backup* é usada em casos de emergência. O extrator de mensagens irá analisar a mensagem e transformá-la no formato apropriado. O gerador de mensagens irá exibir a mensagem para o usuário, tornando-a o mais legível possível. A base de conhecimento armazena as informações de negociações feitas anteriormente.

As estratégias de negociação usada pelo sistema de agendamento de reuniões são:

1. Primeiro que entra, primeiro que sai (*first come, first served, ou first in first out*);
2. maior *rank*;
3. estratégia de votação.

O principal objetivo de cada estratégia é adequar-se às preferências do usuário. Na primeira estratégia, quando um agente recebe uma mensagem de um compromisso, ele é alocado caso esse horário esteja

disponível em sua agenda. A estratégia de maior *rank* é usada para resolver conflitos quando uma mensagem é recebida e ocorre o choque de horários, então o agente deve analisar as preferências do usuário e decidir qual agente tem mais prioridade.

A estratégia de votação é a última a ser usada, sendo apenas requisitada em casos em que a negociação entre agentes atingiu o limite máximo. Nessa estratégia o agente que sugeriu a reunião irá procurar em sua agenda os grupos de horários em que a reunião poderá acontecer. Após a definição dos grupos, uma mensagem é enviada para o servidor e então essa mensagem é redirecionada para os outros agentes. Uma votação é realizada e cada agente deve votar em um determinado horário. O horário mais votado é o escolhido para a reunião. Em casos de empate, o servidor irá escolher o horário que possuir maior prioridade. A prioridade de cada participante da reunião, que é definida pelas preferências do usuário, é somada e então obtém-se a prioridade do compromisso.

A implementação de um protótipo do sistema foi feita utilizando a linguagem Java. Como mencionado anteriormente, o paradigma cliente-servidor foi utilizado para desenvolver a arquitetura. A implementação do servidor foi feita de modo que todos os agentes estejam conectados. A comunicação entre os agentes utilizou *Knowledge Query Manipulation Language (KQML)*.

Os exemplos para validação do sistema focaram em mostrar como os agentes interagem entre si e as mensagens envolvidas no processo de negociação. Foram usados quatro agentes, sendo cada agente representante de um usuário.

2.6.3 Towards host-to-host meeting scheduling negotiation

Megasari et al. (2015) apresentam uma nova abordagem de negociação em agendamento de reuniões. Onde um agente representa um grupo de indivíduos, sendo que ele utiliza informações incertas para representar as preferências do usuário. Os possíveis participantes de uma reunião tem suas informações centralizadas em um único ponto, tornando possível uma melhoria na definição da preferência dos usuários. Essa abordagem é denominada de negociação entre *Host-to-Host*.

Uma adaptação do mecanismo da taxa de Clark foi utilizado para auxiliar na definição das preferências de um conjunto de pessoas. A taxa de Clark visa garantir que todos os participantes de uma determinada atividade consigam atingir um certo equilíbrio, de forma que

não exista alguma espécie de prejuízo.

O mecanismo de negociação apoia-se na teoria de gerenciamento de conflitos e negociação em sistemas multiagente. Ambos estão fortemente associados com a densidade de calendário. A densidade de calendário é uma das variáveis usadas para medir a complexidade do agendamento de uma reunião e ela pode ser calculada a partir da razão entre o número de horários de uma agenda e o número de horários disponíveis. Dada a densidade de um calendário, é possível definir qual a melhor estratégia para a resolução de conflitos. Esse valor também é disponibilizado para todos os outros agentes, ou seja, é armazenado em um calendário global.

A teoria de gerenciamento de conflitos tem como função criar um equilíbrio através de ações em determinadas situações, as ações em questão são: competição, prevenção, acomodação, colaboração e comprometimento. Esse trabalho utiliza duas estratégias para a resolução de conflitos: criação de valor e reivindicação de valor. A estratégia de criação de valor possui ações mais voltadas para prevenção, comprometimento e colaboração. Já a estratégia de reivindicação está atrelada a acomodação, comprometimento e competição. A estratégia usada para a negociação deve considerar as vantagens e custos de cada estratégia e suas consequências.

Conforme abordado anteriormente, o mecanismo de negociação utilizado usa duas estratégias, sendo elas adaptações da criação de valor e reivindicação de valor. Na estratégia baseada na criação de valor, o agente tende a conceder horários na etapa de negociação, enquanto que na estratégia de reivindicação de valor o agente terá um comportamento voltado a imposição de horários. Em uma reunião onde a prioridade é alta, a estratégia comumente adotada é a de impor horários, enquanto que na de baixa prioridade o horário geralmente é negociado através da alteração de preferências.

Apesar de existir a possibilidade de usar informações referentes a preferências e histórico de reuniões para auxiliar no processo de negociação. O que é feito é a transferência de informações do calendário global para cada agente, onde as informações transferidas são referentes a reuniões agendadas de outros agentes.

A prioridade para cada reunião é definida por meio de valores que variam de zero à nove. Sendo o valor zero atribuído a horários em que não se deseja ter a reunião em negociação, já o valor nove está atrelado a preferência pela ocorrência da reunião naquele horário.

Para adaptar a teoria de gerenciamento de conflitos, foi definido que a assertividade seria usada como uma ação realizada. A coopera-

tividade foi abordada com ações que possibilitam a negociação. Com isso os agentes conseguem fazer uso dos dois modos mencionados anteriormente, o que possibilita a resolução de conflitos em agendamento de compromissos. Já a relação da densidade de calendário é controlada através do mecanismo de taxa de Clark, que visa garantir um bem estar social.

O trabalho não apresenta nenhum cenário para testes do modelo proposto, sendo esse um dos objetivos de trabalhos futuros.

2.6.4 Automated negotiation for e-commerce decision making: A goal deliberated agent architecture for multi-strategy selection

Nesse trabalho, Cao et al. (2015) têm como principal objetivo construir um agente que seja capaz de usar diversas técnicas de negociação conforme o cenário apresentado a ele. O autor menciona a "brecha" que existe entre estratégias de negociação e a suas aplicações no mundo real. Isso ocorre devido ao foco estar no uso de apenas uma estratégia, de forma que em casos de imprevisibilidade isso se tornaria um empecilho.

A arquitetura conceitual do agente está baseada no modelo *Belief-Goal-Plan* (BGP), que é uma extensão do modelo clássico BDI. Segundo Cao et al. (2015), a crença de um agente é o entendimento do agente sobre o que está sendo negociado. Isso abrange propriedades relacionadas ao conhecimento do domínio, parâmetros do ambiente, as crenças de outro agente e etc. A crença de um agente é fundamental para a tomada de decisão em negociações, já que é algo que sofre diversas alterações até se chegar em uma decisão. A definição de uma crença de negociação pode ser expressa como $B = \langle I, R, S, M \rangle$, onde:

- I representa a crença desencadeada por alguma interação, seja ela do ambiente ou de um outro agente que esteja envolvido na negociação;
- R representa as crenças em tempo de execução, que são obtidas quando o agente executa um raciocínio. Essas crenças podem ser alteradas conforme o tempo;
- $S = \{s_1, s_2, s_n\}$ é o conjunto de estratégias de negociação disponíveis para serem usadas por um agente. Sendo s_1 uma função $I \rightarrow O$, onde I é uma crença gerada por alguma proposta de negociação e O é a oferta que será feita em seguida.

- $M = \{m_1, m_2, m_n\}$ é conjunto de modelos de utilidades, que é formado por diversas funções de utilidades, sendo cada uma relacionada com a estratégia que será adotada na negociação.

Um agente baseado na arquitetura BGP é definido pelo autor como uma tupla $\langle B, G, P, C, \text{ouvir}, \text{deliberar}, \text{plano}, \text{reação} \rangle$ onde:

1. B é a crença de negociação do agente;
2. G é o conjunto de objetivos do agente. Que representa o estado de mundo almejado por ele;
3. $P = \{p_1, p_2, p_n\}$ é o conjunto de planos de negociação que envolvem ações concretas que um agente irá realizar para atingir seus objetivos;
4. $C = \{c_1, c_2, c_n\}$ é o conjunto de capacidades de um agente. É o módulo do agente que contém crenças, objetivos e planos, que podem ser salvos e reusados em algum outro momento;
5. ouvir: $\alpha(B) \times GS \times LS \rightarrow \alpha(B)$ é uma função que representa as novas crenças geradas a partir de um estado global (GS) e estado local (LS).
6. deliberar: $\alpha(B) \times \alpha(G) \rightarrow \alpha(G)$ é a função responsável por descrever o processo de deliberação de uma negociação. Através do uso de um conjunto de crenças e dos objetivos atuais, novos objetivos são gerados;
7. plano: $\alpha(B) \times \alpha(G) \rightarrow \alpha(P)$ é a função que planeja quais ações serão tomadas na negociação;
8. reação: $GS \times LS \rightarrow \alpha(P)$ essa função é usada em situações onde a etapa de deliberação não é efetuada, momento em que o agente apenas responde às mudanças do ambiente.

As principais etapas envolvidas na negociação e a arquitetura do sistema do agente podem ser vistas na figura 3.

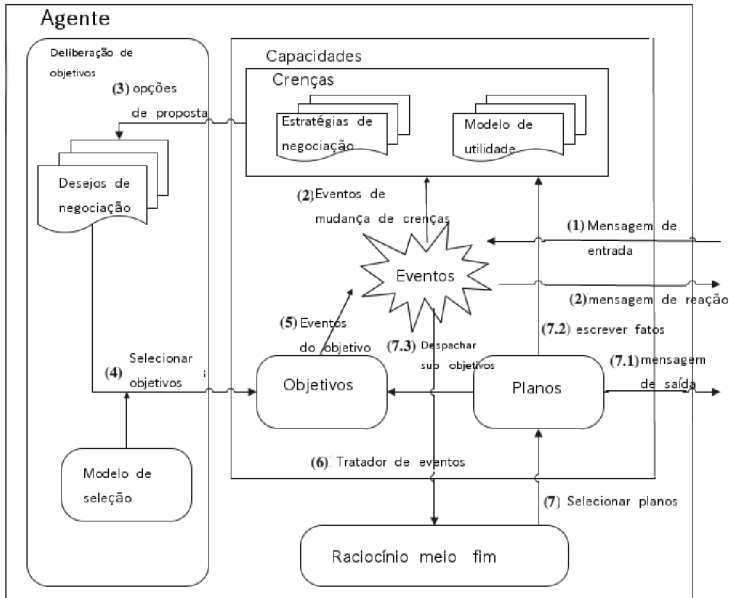


Figura 3 Arquitetura BGP, adaptado de Cao et al. (2015).

O autor propõe que na existência de diversos atributos em algo que está negociando, muito comum em vendas online, a seleção de estratégia para negociação está atrelada ao fato da ocorrência de concessões por parte dos agentes. Quando não for possível achar um equilíbrio na negociação de um atributo, o agente deve empregar regras de concessão. Com isso é possível a ocorrência da negociação de atributos de forma individual, tornando-se uma negociação bilateral, ou seja, um único atributo é negociado durante o processo.

Os testes do modelo de agente e sua arquitetura foram feitos em um caso clássico de negociação entre vendedor e cliente em que apenas um atributo era o foco da interação. A negociação ocorreu entre softwares, deixando para futuros trabalhos a interação entre humano e software.

Ao desenvolver um protótipo do protocolo para agendamento automático de reuniões, alguns aspectos podem ser associados com os trabalhos abordados nessa seção. Existe uma preocupação em representar as preferências do usuário e também a maneira como eles negociam.

Enquanto a maior parte dos trabalhos abordados associam as preferências dos usuários às pessoas. O protocolo apresentado no pró-

ximo capítulo define prioridades para os assuntos, horários e também para pessoas.

As estratégias de negociação propostas pelos autores diferenciam-se em diversos aspectos do que é realizado nessa pesquisa. Hossain e Shakshuki (2013) apresentam um cenário onde agentes tomam suas decisões baseando-se nas respostas dos agentes que representam uma importância maior. Shakshuki et al. (2008) utiliza três estratégias de negociação, onde cada uma é usada em momentos distintos da negociação. Megasari et al. (2015) desenvolvem suas estratégias focando na busca pelo equilíbrio definido pela taxa de Clark. Cao et al. (2015) defendem que a existência de um mecanismo para selecionar uma determinada estratégia em diferentes momentos da negociação.

O protocolo apresentado no capítulo 3 utiliza duas estratégias de negociação em diferentes momentos, próximo ao que foi mostrado por Cao et al. (2015), Megasari et al. (2015), porém não tem seu foco em taxa de Clark. Diferente do que foi apresentado no cenário do trabalho de Hossain e Shakshuki (2013), o agente não necessita ter acesso às respostas dos outros agentes para sua tomada de decisão. Enquanto que Shakshuki et al. (2008) utiliza agentes que representam um grupo de pessoas, o protocolo desenvolvido representa apenas uma pessoa.

3 DESENVOLVIMENTO

Em agendamento automático de reuniões um agente representa um usuário e comunica-se com outros agentes para definir um horário para compromissos (HOSSAIN; SHAKSHUKI, 2013). Nesse trabalho será assumido que um compromisso é uma atividade que envolve no mínimo duas pessoas e tem um assunto a ser tratado, ou seja, pode-se definir o compromisso como uma reunião.

Com base nos trabalhos abordados na seção 2.6, nos mecanismos de negociação e interações em SMA mostrados nas seções 2.5 e 2.2, nas próximas seções apresenta-se uma proposta de um protocolo de negociação em sistemas multiagente para agendamento automático de reuniões.

A seção 3.1 apresenta os principais aspectos empregados no desenvolvimento do protocolo. Na seção 3.2 é mostrado através de pseudo-algoritmos o funcionamento do protocolo e a maneira como as alocações de reuniões são realizadas. Já no capítulo 4 são abordadas decisões de implementação e os cenários utilizados para testar o protótipo do protocolo.

3.1 PROPOSTA DE UM PROTOCOLO DE NEGOCIAÇÃO EM SISTEMAS MULTIAGENTE

Para atingir o objetivo de acoplar ao agente um módulo encarregado de alocar horários para reuniões é necessário o uso de estratégias de negociação, de forma que seja possível analisar o seu funcionamento e a sua aplicabilidade em arquiteturas de agentes já existentes.

É válido ressaltar que o protocolo desenvolvido aplica-se à agentes que tenham algum mecanismo de raciocínio e capacidade de colaboração com outros agentes. Sendo assim, agentes reativos estão fora do escopo do protocolo proposto nesse trabalho.

Tendo em vista os pontos abordados anteriormente, foi implementado o protótipo de um protocolo de negociação para SMA em agendamento automático de compromissos. A metodologia utilizada para descrever o protocolo é a *STRATUM*, sua descrição encontra-se no anexo 6.1.

Essa metodologia está baseada em 5 etapas:

1. Especificação dos objetivos do agente;

2. Descrição das capacidades do agente;
3. Construção do ambiente em que o agente está inserido;
4. Construção de estratégias e táticas;
5. Teste e refinamento.

Mesmo a metodologia *STRATUM* sendo criada para agentes que utilizam heurísticas para negociar, Rahwan et al. (2007) deixam claro que nem sempre é possível, ou necessário, seguir todos passos de forma fiel. Sendo assim, esse trabalho utiliza essa metodologia com foco na descrição do protocolo.

3.1.1 Especificação dos objetivos do agente

Os objetivos do agente em agendamento automático de compromissos podem ser divididos em:

1. Alocar uma reunião;
2. Alocar uma reunião em um horário que satisfaça as preferências do usuário;
3. Sua proposta de reunião deve ser aceita pela maior quantidade possível de envolvidos. Ou seja, o agente deve ser flexível a ponto de analisar respostas e definir o melhor horário para o grupo.

3.1.2 Descrição das capacidades do agente

Para entrar em uma negociação todo agente deve ser capaz de interagir com outros agentes que estão disputando um mesmo recurso (JENNINGS et al., 2001). A etapa de descrição das capacidades do agente está voltada para a definição de quais capacidades o agente utiliza durante a definição de compromissos (RAHWAN et al., 2007). A tabela 2 descreve as capacidades de um agente em um AAR (Agendamento Automático de Reuniões).

As capacidades que utilizam acessos à agenda são realizadas por meio de acessos à base de crenças que será apresentada na subseção 3.1.3. As capacidades C4 e C5 necessitam de um aprofundamento maior, já que elas são essenciais para a definição das estratégias usadas

durante as negociações. A capacidade C4 será abordada na subseção 3.1.5 e a capacidade C5 na subseção 3.1.6.

Tabela 2 – Definição das capacidades dos agentes retirada de (RAHWAN et al., 2007).

	Capacidade	Descrição
C1	Realizar proposta	Definição de um horário, assunto e agentes envolvidos para criar uma proposta.
C2	Aceitar proposta	-
C3	Rejeitar proposta	-
C4	Criar <i>feedback</i>	Criação de uma resposta, que contenha os horários disponíveis para aquele compromisso.
C5	Definir esforço	Todo horário de uma proposta pode ser associado a um valor numérico, que representa o esforço necessário para alocar esse horário.
C6	Criar resposta	Definir uma resposta, utilizando C4 e C5.
C7	Analisar <i>feedback</i>	Ao receber uma resposta com um compromisso, o anfitrião analisa cada uma, de forma a definir o melhor horário em relação ao conjunto de respostas.
C8	Analisar respostas que contenham o esforço	Anfitrião analisa cada resposta, e armazena a soma do esforço e o horário daquela rodada.
C9	Acessar agenda	Buscar e atualizar informações sobre horários de compromissos.
C10	Acessar preferências do usuário	Buscar informações relacionadas às preferências do usuário.

3.1.3 Base de crenças

O funcionamento do protocolo encontra-se na consulta à base de crenças de um agente, esse banco de crenças contém informações relevantes sobre as preferências do usuário que é representado pelo agente,

tais como:

- prioridade de envolvidos: quais são as prioridades das pessoas ou organizações que pretende-se agendar um compromisso. Um professor tende a atribuir uma prioridade maior a um orientando do que a um aluno que não esteja envolvido em alguma pesquisa que esteja ligado a ele;
- prioridade de assuntos: semelhante a prioridade de envolvidos, onde um assunto possui um valor que represente sua prioridade na agenda;
- prioridade de horários: definição de quais são os horários em que se deseja agendar um compromisso;
- horários disponíveis e indisponíveis;
- envolvidos em um assunto.

Toda prioridade é definida por valores inteiros que estão entre 1 e 10, onde o valor 1 representa o valor mais prioritário e o valor 10 representa o valor com menor prioridade. A definição desses valores é uma tentativa de reproduzir as preferências do usuário, ou seja, se um usuário deseja ter suas reuniões marcadas sempre no período da manhã, então é necessário que o agente tenha os horários da parte da manhã com a prioridade contendo o valor 1. O mesmo vale para a prioridade de pessoas e assuntos.

3.1.4 Definindo prioridade de assuntos e pessoas

Como o objetivo do trabalho é a construção de um protocolo de negociação em SMA para o agendamento automático de reuniões, a maneira como as prioridades são representadas estão simplificadas.

A limitação na alocação de compromissos está atrelada ao calendário do usuário representado pelo agente e os assuntos ainda não tratados em nenhuma reunião. Caso um agente queira iniciar uma proposta de reunião, ele deverá verificar se existem assuntos que ainda não foram tema de nenhuma reunião, os horários disponíveis também influenciam tanto na proposta quanto na análise propostas.

As atualizações de horários e status de assuntos desconsideram diversos fatores presentes na revisão de crenças. Sendo assim, fica fora do escopo desse trabalho abordar aspectos de revisão de crenças, como

por exemplo os postulados AGM (Alchourrón Gärdenfors e Makinson) abordados no trabalho de Peppas (2008).

Assim como no mundo real, agentes também tem suas crenças alteradas com o passar do tempo. A alteração de crenças relacionadas a prioridade de pessoas, assuntos e horários durante a negociação podem sofrer decréscimos ou acréscimos, de forma a tornar esses aspectos mais prioritários ou menos prioritários, respectivamente.

Outro ponto importante, é que a implementação do protótipo do protocolo foi feita de forma que seria relativamente simples adicionar algum mecanismo para definir as prioridades. Já que bastaria alterar os fatos e estruturas que estão associados às prioridades.

3.1.5 Resposta através do *feedback*

O rumo de uma negociação é definido pela distância que a proposta atual está da área de aceitação do outro agente, ou seja, o agente ofertante deve sempre formular sua próxima proposta para que ela esteja contida na área de aceitação do outro agente (JENNINGS et al., 2001). Uma maneira de encurtar essa distância é revelar partes da área de aceitação do agente envolvido na negociação, de forma que a resposta indique o quão longe ou quão perto a proposta se encontra.

No protocolo proposto essa tarefa pode ser feita através de uma resposta que contenha os horários em que o agente esteja disponível. Mandar uma mensagem com todos os horários disponíveis poderia causar uma sobrecarga na troca de mensagens, então é fundamental utilizar algum mecanismo que seja capaz de identificar quais são os horários apropriados para um determinado compromisso.

Para contornar os problemas identificados anteriormente, é necessário usar as informações que representam as preferências do usuário para estipular os melhores horários. A geração do *feedback* visa associar os horários disponíveis com as informações referentes a prioridade do assunto, do anfitrião (agente que enviou a proposta de reunião) e dos envolvidos. Ao receber uma mensagem enviada por um agente com prioridade 1, todos os horários disponíveis com prioridade 1 serão anexadas na resposta.

3.1.6 Definição de esforço para alocação de um compromisso

O uso de leilões em sistemas multiagente é um mecanismo muito utilizado para decidir qual agente irá utilizar um certo recurso. Apoiando-se nesses mecanismos e suas derivações, conforme mostrado na seção 3.1, o protocolo proposto define uma métrica para determinar o esforço que seria necessário para alocar um horário. Essa etapa de definição da métrica ocorre na análise de uma proposta e sua definição pode ser obtida por meio dos seguintes valores:

- ps é o valor da prioridade do assunto;
- pr é o valor da prioridade do anfitrião;
- pg é a média da prioridade dos envolvidos;
- h horário proposto;
- O valor numérico que representa a rodada em que se encontra a negociação também tem influência na definição da métrica de esforço;
- Sendo $H = \{h_1, h_2, \dots, h_n\}$ o conjunto dos horários disponíveis que possuem prioridades pr e $pr + rodada$. O conjunto $D = \{|h_1 - h|, |h_2 - h|, \dots, |h_n - h|\}$ representa o módulo da diferença entre os horários disponíveis e o horário proposto. Onde $\min(di) \mid di \in D \text{ e } di < di'$, ou seja, di representa a menor diferença entre o horário disponível e o horário proposto.

Supondo que um agente receba uma mensagem que propõe um compromisso no dia 05/09/2016 às 7:30. Após consultar sua base de crenças e obter o valor 1 como prioridade do anfitrião e prioridade do assunto, o agente irá consultar à base de crenças o horário com prioridade 1 e valor 2 (prioridade do anfitrião somada com a rodada da negociação). Ao finalizar a busca o agente obtém os seguintes horários do dia 05/09/2016: 7:00, 8:00 e 8:30. A diferença entre o horário proposto e os horários obtidos são: 0:30, 1:00 e 1:30. Sendo assim, o horário que representa a menor diferença é o das 7:00.

O esforço pode ser definido como: $e = ps + pr + pg + n$. Onde n é o valor normalizado da menor diferença, variando de 0 à 100. A maneira como a normalização é realizada pode ser vista no pseudo-algoritmo 4.

Essa diferença tem como principal objetivo garantir que horários muito distantes não sejam alocados para um compromisso, tendo

em vista que quanto mais distante for marcada uma reunião, maior a chance de ocorrer algum imprevisto, o que acarretaria em um cancelamento.

Ao receber um conjunto de respostas o anfitrião leva em consideração a prioridade dos envolvidos combinada com o valor do esforço, dando preferência aos envolvidos que representam uma maior importância para o anfitrião. Todas essas abordagens servem para aproximar as áreas de aceitação dos envolvidos.

Através do cenário abaixo é possível notar algumas limitações ao adotar essa estratégia para a negociação:

- O anfitrião envia uma proposta para n pessoas que ele gostaria que participasse da reunião;
- Cada envolvido define o esforço daquele compromisso, ou seja, quanto maior o valor, maior será o esforço;
- Ao receber n respostas, o anfitrião depara-se apenas com o esforço de um horário, o que acarreta uma falta de informação, já que o agente tem informações apenas de um horário, o que impossibilita definir uma direção na procura pela intersecção da área de aceitação dos envolvidos.

Para contornar a limitação mostrada no cenário acima pode-se repetir essa proposta para outros horários e ao fim da negociação definir qual seria o horário que requer menor esforço. Isso impede que a negociação dure por um tempo muito prolongado.

Mesmo sendo limitada, a simplicidade da métrica do esforço a torna um mecanismo de estratégia que pode ser usada em conjunto com outras técnicas. Ela pode ser usada em casos em que dois horários estão disponíveis por todos os envolvidos e então o anfitrião pode escolher qual o horário requer um menor esforço, aumentando assim a chance de todos aceitarem.

3.1.7 Construção do ambiente

Nas seções anteriores foram apresentadas as principais características de cada capacidade do agente. Essa subseção irá abordar a etapa 3 do *STRATUM*.

A etapa 3 tem como objetivo descrever em quais situações o agente será exposto durante a negociação, sendo um passo importante para definir o funcionamento do protocolo (RAHWAN et al., 2007). Nessa

etapa são mostradas algumas características da comunicação utilizada e características presentes em cada mensagem.

Conforme abordado na seção anterior, o protocolo de negociação faz o uso de duas estratégias: geração de *feedback* e uma métrica de esforço, sempre acessando a base de crenças para obter informações referentes a preferências do usuário. Essas características definem dois aspectos importantes da comunicação entre os agentes, sendo elas:

- O ambiente em que os agentes estão inseridos é considerado informativo em relação a quem propôs um compromisso e quem respondeu a proposta. Entretanto, envolvidos não comunicam-se entre si, de forma que apenas o anfitrião tem domínio das informações das respostas de todos os envolvidos;
- Assim como na linguagem de comunicação FIPA que requer sinceridade (RAHWAN et al., 2007). O protocolo em questão também segue esse padrão, pois ao enviar uma proposta ou responder à proposta, o agente sempre irá anexar informações verdadeiras.

3.2 ESTRATÉGIAS E FUNCIONAMENTO DO PROTOCOLO

A etapa 4 do *STRATUM* visa definir as estratégias utilizadas na negociação. As próximas subseções apresentam as estratégias adotadas na etapa de negociação e também como o anfitrião analisa cada resposta dos envolvidos, de forma a escolher um horário para a reunião.

Todas as estratégias utilizadas durante a execução do protocolo assumem que o agente está sendo sincero sobre as informação referentes à representação da sua região de aceitação.

3.2.1 Propondo compromissos

Quando deseja-se marcar uma reunião com alguém, algumas perguntas devem ser respondidas: quem estará envolvido nessa compromisso? qual o principal assunto dessa reunião? qual será o horário? qual o grau de importância do assunto e dos envolvidos? Em agendamento de reuniões automático essas questões também estão presentes e devem ser levadas em consideração durante a etapa de negociação.

A proposta de uma reunião ocorre quando existe um assunto na base de crenças que ainda não foi tratado em nenhuma reunião. Ao assumir a existência dessa característica, então pode-se considerar que o agente tem a intenção de marcar uma reunião. Tendo disponível o

assunto, é necessário estabelecer os envolvidos e os horários que estão associados a esse assunto. Essa associação é feita por meio do valor da prioridade do assunto, com isso consulta-se à base de crença para obter o horário que possui prioridade igual à prioridade do assunto. O pseudo-algoritmo 1 retrata as etapas necessárias para criação de uma mensagem de proposta.

Algoritmo 1: Pseudo-algoritmo para criação de propostas

início

```

    par prior_assunto=obter_assunto_nao_resolvido();
    horarios=obter_horarios_por_prioridade(prior_assunto);
    membros=obter_pessoas_envolvidas_assunto(prior_assunto);
    retorna criar_proposta(nome, assunto, horario,
        membros);

```

fim

3.2.1.1 Definindo a resposta

Ao receber a proposta de um compromisso serão obtidas informações das prioridades de quem enviou o convite, de quem possivelmente estará na reunião, o assunto e o horário proposto. A geração da resposta pode ser dividida nos seguintes passos:

1. verificar a disponibilidade do horário proposto: caso o horário proposto esteja disponível, então uma mensagem de aceitação é enviada;
2. definir a prioridade do compromisso: busca-se na base de crenças a prioridade do anfitrião;
3. em casos onde o anfitrião represente um usuário com prioridade baixa, o protocolo leva em consideração a prioridade do grupo. A prioridade do grupo é a média aritmética da prioridade de todos os envolvidos no compromisso;
4. Uma consulta à base de crenças utilizará a prioridade do anfitrião ou a prioridade do grupo para obter os horários com prioridade igual;
5. Casos em que nenhum horário tenha retornado da consulta à base de crenças pode demonstrar que os envolvidos ou o anfitrião não possuam relevância suficiente, porém o assunto pode ser um fator

importante para o usuário. Então uma nova consulta é realizada, porém levando em consideração a prioridade do assunto;

6. Uma mensagem com todos os possíveis horários em que seja possível alocar a reunião é enviada;
7. Caso nenhum horário seja encontrado, uma mensagem recusando o convite será enviada.

Mesmo negando um convite o agente ainda deve participar das próximas iterações da negociação, de forma que ele ainda tenha chances de receber um horário que satisfaça suas condições de agendamento. Os passos descritos acima podem ser transcritos conforme o pseudo-algoritmo 2, onde os comentários representam os passos citados anteriormente.

Algoritmo 2: Pseudo-algoritmo para geração de resposta utilizando *feedback*

Entrada: Mensagem com a proposta

Saída: resposta da proposta

início

```

    se (horario_disponivel(proposta)) então
        retorna gerar_resposta_aceitacao(proposta);
        // item 1
    fim
    prioridade_compromisso=prioridade(anfitriao); // item
    2
    se (prioridade_compromisso >
        PRIORIDADE_MINIMA) então
        prioridade_compromisso=obter_prioridade_grupo();
        // item 3
    fim
    horarios=obter_horarios_prioridade(prioridade_compromisso);
    // item 4
    se (horarios.vazio()) então
        prioridade_assunto=prioridade(assunto);
        horarios=obter_horarios_prioridade(prioridade_assunto);
        // item 5
    fim
    retorna gerar_resposta(horarios); // itens 6 e 7

```

fim

A tabela 3 associa o que foi discutido acima com as capacidades necessárias para realizar a estratégia.

Tabela 3 – Adaptação do modelo de definição de estratégias/táticas do *STRATUM* para o *feedback*.

Nome da estratégia	Geração de <i>feedback</i>
Objetivo	Alocar horário para um compromisso.
Métodos	Informar o resultado do convite, caso o horário esteja ocupado, enviar os horários disponíveis.
Capacidades	C2 e C3 para aceitar ou rejeitar; C9 e C10 para acessar a base de crenças, C4 para criar uma resposta.

3.2.2 Analisando as respostas

Ao receber as respostas da sua proposta o anfitrião deve procurar por um horário que seja aceito por todos os envolvidos. Como nem sempre isso é possível, existem duas estratégias que podem ser adotadas: procurar pelos horários que sejam aceitos pelo maior número de envolvidos ou o horário que seja aceito pelos envolvidos que possuem maior prioridade.

Quando um horário em comum é encontrado, apenas envia-se uma mensagem confirmando o horário e então atualiza-se a base de crenças do anfitrião. Já em casos onde encontrou-se n horários que foram obtidos a partir da procura citada acima, esses horários são anexados em uma mensagem e então enviados novamente para os envolvidos.

3.2.3 Lidando com impasses e conjunto de horários

Ao receberem a nova mensagem com o conjunto de horários cada envolvido deve definir o esforço para o horário contido na mensagem e então enviar a resposta com esse valor. É válido ressaltar que o esforço analisa a distância entre o horário proposto e os horários que possuem prioridade próximas ao compromisso. Com isso tem-se o horário alternativo mais próximo do horário proposto. O pseudo-algoritmo 3 descreve os passos necessários para definir o esforço. É importante no-

tar que na linha 8 do pseudo-algoritmo 3 ocorre uma busca por horário com a soma entre a prioridade e a rodada atual, isso significa que o agente vai tornando-se mais flexível conforme o decorrer da negociação, de forma que horários com prioridades menores sejam utilizados como candidatos.

Assim como na estratégia da geração do *feedback*, aqui também usa-se o modelo de definição de estratégias do *STRATUM*. A definição encontra-se na tabela 4 e apresenta as principais capacidades envolvidas nessa estratégia.

Algoritmo 3: Pseudo-algoritmo para geração da resposta utilizando métrica do esforço.

Entrada: Mensagem com a proposta

Saída: Resposta contendo o esforço

início

```
prioridade_assunto=prioridade(proposta);
prioridade_anfitriao=prioridade(proposta);
prioridade_grupo=prioridade(proposta);
vetor_horarios;
horarios.inserir(obter_horarios_prioridade
(prioridade_anfitriao
+rodada);
horario_maximo = horario_proposto+2 dias;
diferenca=diferenca_horarios(horarios,
    horario_proposto, horario_maximo);
esforco=prioridade_anfitriao + prioridade_assunto +
    prioridade_grupo + diferenca;
retorna criar_resposta_esforco(esforco);
```

fim

Quando o anfitrião receber essas respostas ele deve procurar pelo horário que representa o menor esforço. Essa procura pela menor soma de esforço serve tanto para casos em que nenhum horário foi definido, quanto para existência de dois ou mais horários aceitos por todos os envolvidos. A negociação pode ser finalizada em dois casos:

- Quando o anfitrião receber as respostas com o *feedback*, ele detecta um horário comum entre os participantes;
- A quantidade máximas de rodadas foi atingida e então obtém-se o horário que representa o menor esforço de todos os participantes.

Algoritmo 4: Pseudo-algoritmo para obter a menor diferença entre horários.

Entrada: horários, horario_proposto, horario_maximo

Saída: valor do esforço normalizado entre 0 à 100.

início

 menor_diferenca=obter_menor_diferenca_segundos(horarios,
 horario_maximo);

 valor=menor_diferenca/(horario_maximo.segundos()-
 horario_proposto.segundos());

retorna valor*100;

fim

Tabela 4 – Adaptação do modelo de definição de estratégias/táticas do *STRATUM* para o esforço

Nome da estratégia	Métrica do esforço
Objetivo	Alocar horário para um compromisso.
Métodos	Definir o esforço necessário para agendar o compromisso, no horário proposto.
Capacidades	C9 e C10 para acessar a base de crenças; C5 para definir o esforço.

A figura 4 aborda uma execução em alto nível do protocolo através de um diagrama de interação do *Agent* UML.

Visando facilitar a explicação dos detalhes de funcionamento do protocolo, a próxima seção tem como objetivo abordar detalhes de implementação do protótipo do protocolo mostrado nessa seção.

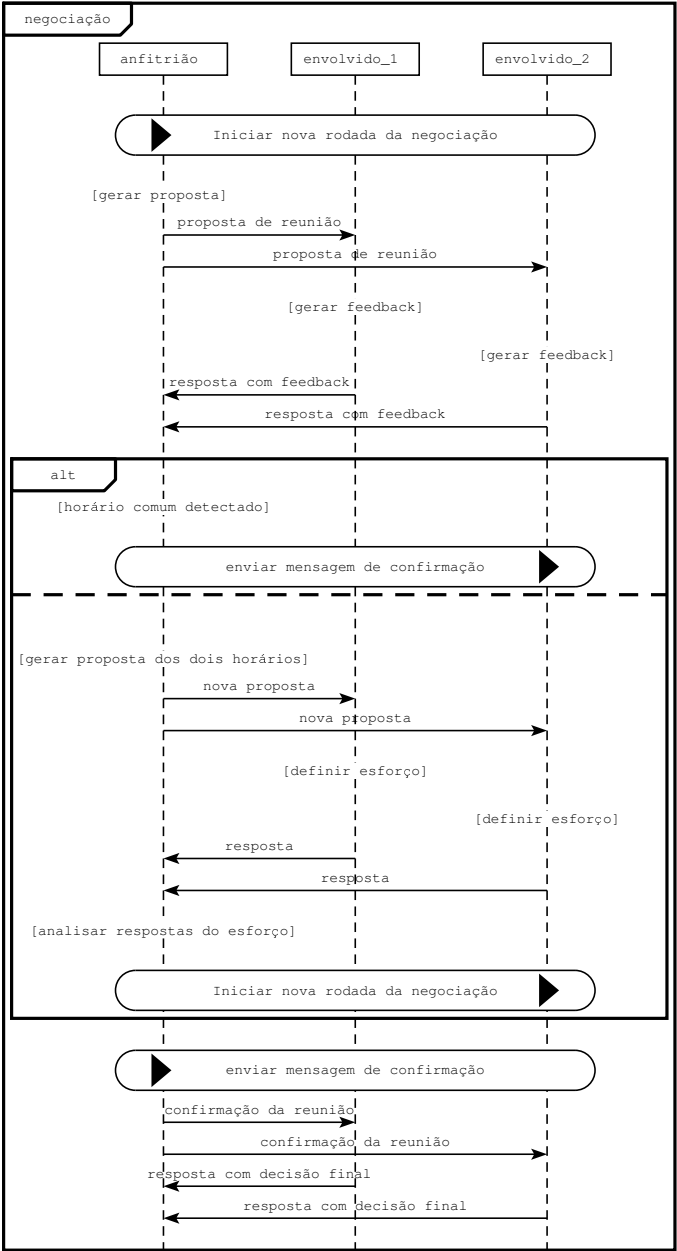


Figura 4 – Diagrama de interação da execução do protocolo de negociação.

4 IMPLEMENTAÇÃO DO PROTÓTIPO DO PROTOCOLO

Essa seção tem como principal objetivo mostrar as etapas necessárias para a implementação do protocolo discutido na seção 3.1. Assim como em 2APL, que utiliza conceitos do Prolog para representar aspectos da sua base de crença (DASTANI, 2008), a implementação do protótipo também usa o Prolog para representar a base de crenças.

A linguagem de programação C++ foi usada para descrever o funcionamento e as etapas do protocolo abordadas na seção 3.1. Por pertencer a um paradigma diferente e sua interação com C++ não ser muito comum, é necessário abordar o Prolog de forma mais detalhada.

4.0.1 Interface C++/Prolog

Prolog é uma linguagem descritiva de propósito geral que está fortemente baseada em lógica de primeira ordem. Um programa em Prolog não é feito através da especificação de um algoritmo como em uma linguagem de programação imperativa. Prolog está voltada para a representação formal de relações e objetos em um problema. Um programa em Prolog preocupa-se na descrição de fatos e relações, deixando de lado a definição de uma sequência de passos para resolver o problema em si (CLOCKSin; MELLISH, 2003).

Nesse trabalho a linguagem Prolog é usada para representar a base de crenças de um agente, sendo que os componentes que estão presentes são: agenda, prioridade de assuntos e pessoas que relacionam-se com o usuário.

O protocolo prevê que cada agente tenha acesso a sua agenda e não consiga realizar nenhum tipo de consulta à base de crenças de outros agentes. Para poder negociar com outros agentes é fundamental que o acesso à base de crenças seja feito a cada iteração da etapa de negociação, isso ocorre por meio de consultas ao Prolog, o que torna necessário o uso de uma interface entre a linguagem de programação C++ e o Prolog.

Para realizar a ligação entre as duas linguagens foi utilizado o SWI-Prolog e a implementação de uma interface entre as linguagens. SWI-Prolog é uma implementação da linguagem Prolog que tem seu desenvolvimento baseado no uso do Prolog com outras linguagens de programação (WIELEMAKER et al., 2012).

A interface C++/Prolog usada nesse trabalho também pertence a SWI-Prolog e foi construída a partir da interface C/Prolog. Apesar de já existir a possibilidade de utilizar a interface C/Prolog em C++, a interface C++/Prolog é considerada muito mais concisa e de fácil uso. Ela possibilita conversão automática de tipos de dados nativos da linguagem C, geração de exceções, realização de consultas ao Prolog e uso de predicados estrangeiros, que é uma maneira de representar um predicado em Prolog através do uso de funções em C++ (WIELEMAKER et al., 2012).

4.0.2 Organização da implementação do protótipo do protocolo

Para implementar as funcionalidades descritas na seção 3.1 foram necessários quatro módulos, sendo eles listados abaixo:

1. desenvolvimento da base de crenças;
2. acessos à base de crenças utilizando C++;
3. definição da estratégia utilizada;
4. definição dos tipos de mensagens presentes no protocolo.

Nas próximas sub-subseções serão apresentados detalhes do desenvolvimento e funcionamento de cada módulo abordado anteriormente.

4.0.2.1 Desenvolvimento da base de crenças

Conforme mostrado em seções anteriores, a base de crenças é encarregada de representar informações relevantes para o agendamento automático de compromissos e conseqüentemente essenciais para a etapa de negociação. Um horário pode ser representado por meio da tupla $H \langle data, dia da semana, horário, prioridade do horário, disponibilidade \rangle$. A disponibilidade e a prioridade do horário na tupla são fundamentais para que ocorra a compatibilidade entre prioridade do anfitrião, envolvidos e assuntos.

Conforme abordado na subseção 3.1.4, existem diferentes maneiras de definir as prioridades e também como elas podem ser alteradas com o passar do tempo. Para fins de testes, foram criados fatos para representar assuntos, pessoas e relacioná-los com suas prioridades. É

válido lembrar que casos em que na base de crenças exista o assunto "Trabalho de conclusão de curso" e que ao receber uma mensagem enviada pelo Orientador com o tema "TCC", não é possível identificar a relação entre as duas. Isso não foi implementado pelo fato de que sua implementação ou não, não influenciaria nos testes necessários para avaliação do protocolo.

Fatos e estruturas utilizados para representar aspectos da agenda podem ser vistos na figura 5.

```
/*estruturas que reresentam um horário*/
horario(data(2016, 09, 05), semana(segunda-feira), hora(7, 30), 1, disponivel).
horario(data(2016, 09, 05), semana(segunda-feira), hora(8, 00), 1, disponivel).
horario(data(2016, 09, 05), semana(segunda-feira), hora(8, 30), 2, disponivel).
horario(data(2016, 09, 05), semana(segunda-feira), hora(9, 00), 2, disponivel).
horario(data(2016, 09, 05), semana(segunda-feira), hora(9, 30), 4, disponivel).
horario(data(2016, 09, 05), semana(segunda-feira), hora(10, 00), 4, disponivel).

/*prioridade de pessoas e assuntos*/
pessoa_prioridade(ana, 1).
pessoa_prioridade(pedro, 2).

assunto_prioridade(agentes, 1).

/*status de um assunto e pessoas envolvidas em um assunto*/
assunto_status(agentes, pendente).

assunto_pessoa(agentes, ana).
assunto_pessoa(agentes, pedro).
```

Figura 5 – Fatos e estruturas da base de crenças.

Apesar de existirem predicados encarregados de verificar horários disponíveis, assuntos não resolvidos e outros aspectos, seu uso será explorado na seção seguinte, tornando mais clara o uso da interface entre C++/Prolog.

4.0.2.2 Acessos à base de crenças utilizando C++

Esse módulo faz o uso da interface C++/Prolog mostrada na seção 4.0.1, sendo essa uma etapa fundamental para as estratégias que serão aplicadas em negociações de compromissos. Através de consultas à base de crenças é possível tomar uma decisão baseando-se sempre no resultado da consulta.

Ao realizar uma consulta ao Prolog é preciso fazer o uso de termos para estruturar uma consulta. A interface usada oferece o *PlTerm* que representa termos e cada termo é tratado como um objeto da linguagem de programação C++. Todo termo é definido por meio de valores de dados da linguagem C++, por exemplo: para verificar se

um compromisso pode ser alocado em um horário é necessário definir os termos por meio do horário em questão.

Após ter todos os termos definidos é preciso instanciar um objeto que irá conter as variáveis e os termos que serão usados na consulta. Então realiza-se a consulta ao Prolog utilizando o objeto *PlQuery*. O conjunto de resultados será inserido na variável que foi passada como parâmetro à consulta. O pseudo-algoritmo 5 exemplifica o funcionamento de uma consulta utilizando a interface C++/Prolog.

Algoritmo 5: Pseudo-algoritmo para exibir horários com determinada prioridade.

```

Entrada: valor da prioridade
PlTermv av(8);
PlTerm dia, mes, ano, hora, minutos, dia_da_semana;
av[0] = ano;
av[1] = mes;
av[2] = dia;
av[3] = hora;
av[4] = minutos;
av[5] = dia_da_semana;
av[6] = prioridade;
PlQuery q("prioridade_horarios", av);
repita
|   print av;
até (q.next_solution());

```

Mesmo sendo uma das últimas etapas da negociação a atualização da agenda é essencial, já que é ela quem apresenta os resultados ao usuário. Prolog oferece dois mecanismos para adicionar e remover fatos e predicados da base de dados, o *assertz* e *retract*. Do ponto de vista de implementação, um horário nunca será removido e sim atualizado, de modo que apenas sua disponibilidade é alterada. Sendo assim, é preciso executar um *retract*, seguido de um *assertz* para tal operação. Da mesma forma que foi mostrada no pseudo-algoritmo 5, a invocação de um outro predicado pode ser feita de maneira semelhante para atualizar um horário.

4.0.3 Definição da estratégia utilizada

Conforme mostrado na seção 3.1, existem duas estratégias que podem ser utilizadas na negociação, a geração do *feedback* e o uso da métrica de esforço. O esforço pode ser associado com uma espécie de votação, onde envolvidos respondem a proposta com um valor numérico, que representa sua "vontade" de que o compromisso seja alocado no horário proposto. A classe *Negotiation_Strategy* aborda ambas as implementações e as etapas necessárias.

Uma maneira de representar o protocolo desenvolvido nesse trabalho é assemelhando cada rodada da negociação com uma determinada técnica de negociação, ou seja, em uma rodada responde-se a proposta usando a métrica de esforço e na outra gera-se um *feedback* como resposta. O pseudo-algoritmo 6 exemplifica tal característica.

Ao receber a proposta para agendar um compromisso, é papel da classe *Negotiation_Protocol* decidir qual estratégia deverá ser usada. Se a mensagem que foi recebida representar uma iteração em que a resposta seja gerada utilizando *feedback*, um método que representa o pseudo-algoritmo 2 será executado. Caso contrário, a resposta conterá o valor do esforço, conforme apresentado no pseudo-algoritmo 3.

A classe *Negotiation_Protocol* também tem influência sobre o que o anfitrião deve fazer ao receber as respostas da proposta. De forma análoga, a análise das respostas também é definida pela rodada da negociação. As duas formas são implementações baseadas nos pseudos-algoritmos 3 e 2.

Quando um anfitrião detecta que houve um consenso entre as respostas ou que o máximo de iterações foi atingida, as iterações são finalizadas e então uma mensagem final com o horário decidido é enviada aos envolvidos. O pseudo-algoritmo 6 evidencia os passos apresentados na figura 4.

4.0.4 Tipos de mensagens presentes no protocolo

Ao empregar o uso de alguma estratégia de negociação é essencial que as mensagens utilizadas sejam consistentes. Assim como outros protocolos de comunicação, existem diversos tipos de mensagens, porém no protocolo proposto toda mensagem trocada entre os agentes deve possuir: o anfitrião do compromisso, o principal assunto, o horário e os envolvidos. Os tipos de mensagens existentes no protocolo são:

Algoritmo 6: Pseudo-algoritmo com a execução do protocolo.

início

```

    Estrategia_de_Negociacao ns("nome_agente");
    mensagem = ns.gerar_proposta();
    membros = mensagem.participantes();
    vetor respostas;
    repita
        para (cada  $m \in \text{membros}$ ) faça
            resposta =
                m.gerar_resposta_prioridade(mensagem);
            respostas.inserir(resposta);
        fim
        nova_proposta=ns.analisar_respostas_feedback(respostas);
        /* nova mensagem contém dois horários: o
           que representa a maioria dos envolvidos,
           e aquele em que a prioridade dos
           envolvidos seja a maior possível.          */
        respostas.limpar();
        se (!ns.confirmacao_detectada()) então
            para (cada  $m \in \text{membros}$ ) faça
                resposta=m.gerar_resposta_esforco(nova_proposta);
                respostas.inserir(resposta);
            fim
            mensagem=ns.analisar_respostas_esforco(respostas);
            respostas.limpar();
        fim
    até (!ns.fim_da_negociacao());
    enviar_proposta_final(mensagem);

```

fim

- Mensagem de proposta: mensagem que dá início às negociações entre agentes, ou seja, a sugestão de um horário aos envolvidos;
- Mensagem de resposta de aceitação: usada em casos em que o horário proposto esteja disponível;
- Mensagem de resposta com *feedback*: quando o horário proposto já está alocado para outra atividade, é enviada como resposta os horários em que existe a possibilidade de ocorrer o agendamento;
- Mensagem de resposta com o esforço: mensagem contendo o remetente e o esforço de um horário;
- Mensagem de confirmação do anfitrião: após o anfitrião obter as respostas de todos os envolvidos, as seguintes situações podem ocorrer: (i) respostas contêm a aceitação de todos os envolvidos, (ii) respostas contendo *feedback* com um horário que represente um consenso ou (iii) a negociação chegou na última rodada. Nesses casos é enviada uma mensagem contendo o horário definido;
- Mensagem de confirmação de presença: depois de ter recebido uma mensagem de confirmação do anfitrião o banco de crenças é atualizado e o horário escolhido é marcado como indisponível. Então é emitida uma mensagem de confirmação de presença ao anfitrião, que também deve atualizar sua base de crenças.

Nessa implementação foi assumido que ao propor um horário, tanto o anfitrião quanto os envolvidos, ao formular uma resposta com *feedback* irão garantir que o horário presente na mensagem não irá mudar de disponível para indisponível até o fim da negociação.

Cancelamentos e reagendamento de reuniões não foram implementadas no protótipo, já que a adição dessas propriedades poderia acarretar em problemas de cascadeamento, onde agentes começam a cancelar e reagendar compromissos, dando início a uma desestruturação das agendas. Entretanto, cancelamento e reagendamento serão abordados em trabalhos futuros, bem como o impacto de suas ocorrências na definição das prioridades.

4.1 CENÁRIOS E TESTES REALIZADOS

Para testar o protocolo apresentado nas seções anteriores foram definidos cenários que abordam as principais características das técnicas utilizadas durante a negociação. Foram usados três cenários, sendo

que o primeiro explora as limitações da métrica de esforço, já o segundo cenário está voltado para as limitações da geração de *feedback* como uma resposta. O terceiro cenário é uma adaptação do cenário apresentado em *A Negotiation Protocol for Meeting Scheduling Agent* e tem o objetivo de comparar os resultados obtidos entre os protocolos.

Nos cenários 1 e 2 serão comparadas a métrica do esforço, a geração do *feedback* e uma combinação da métrica do esforço com o *feedback*. É válido ressaltar que a combinação de ambas as técnicas formam a estratégia utilizada pelos agentes durante a execução do protocolo de negociação.

Já no cenário 3, apenas o protocolo definido por Hossain e Shakshuki (2013) e o protocolo proposto nesse trabalho serão comparados.

4.1.1 Cenário 1

O cenário 1 envolve três agentes que serão identificados como e1, r1 e r2. Sendo que o agente e1 irá enviar uma mensagem propondo um horário para uma reunião, já os agentes r1 e r2 irão interagir com o e1 para tentar definir um horário em comum. Um trecho da agenda de cada um dos agentes envolvidos podem ser vistas nas tabelas 5, 6 e 7.

Tabela 5 – Agenda de e1.

Data	Dia da semana	Horário	Prioridade	Status
05/09/2016	segunda-feira	7:30	1	disponível
05/09/2016	segunda-feira	8:00	1	disponível
05/09/2016	segunda-feira	8:30	1	disponível
05/09/2016	segunda-feira	9:00	1	disponível
05/09/2016	segunda-feira	9:30	2	disponível
05/09/2016	segunda-feira	10:00	2	disponível
05/09/2016	segunda-feira	10:30	2	disponível

Conforme abordado na subseção 3.2.1, o assunto com maior prioridade que ainda não foi resolvido será usado para buscar por um horário na base de crenças. Nesse cenário será utilizado um assunto que possui prioridade 1, sendo esse o assunto não resolvido com maior prioridade. Ao utilizar apenas o esforço como estratégia, os horários propostos a cada iteração serão os seguintes: 7:30, 8:00, 8:30 e 9:00. A quantidade de horários estipula o número máximo de iterações que irão ocorrer na negociação. Já nas outras duas técnicas será usado o

primeiro horário disponível que possua a prioridade adequada.

Tabela 6 – Agenda de r1.

Data	Dia da semana	Horário	Prioridade	Status
05/09/2016	segunda-feira	7:30	1	indisponível
05/09/2016	segunda-feira	8:00	1	indisponível
05/09/2016	segunda-feira	8:30	1	indisponível
05/09/2016	segunda-feira	9:00	1	disponível
05/09/2016	segunda-feira	9:30	2	disponível
05/09/2016	segunda-feira	10:00	2	disponível
05/09/2016	segunda-feira	10:30	2	disponível

Tabela 7 – Agenda de r2.

Data	Dia da semana	Horário	Prioridade	Status
05/09/2016	segunda-feira	7:30	1	indisponível
05/09/2016	segunda-feira	8:00	1	indisponível
05/09/2016	segunda-feira	8:30	1	indisponível
05/09/2016	segunda-feira	9:00	1	disponível
05/09/2016	segunda-feira	9:30	2	disponível
05/09/2016	segunda-feira	10:00	2	disponível
05/09/2016	segunda-feira	10:30	2	disponível

Ao utilizar a métrica do esforço é possível concluir que o único horário possível para o compromisso é o das 9:00, que será proposto na última iteração da negociação, levando 5 rodadas para chegar em um acordo. Isso evidencia algumas limitações que foram descritas na subseção 3.1.6.

Já nas outras duas abordagens é possível notar que o acordo ocorrerá em apenas uma iteração. Conforme mostrado na subseção 3.1.5, ao receber uma proposta com o horário das 7:30, r1 e r2 irão procurar em suas agendas e obter apenas um horário, o das 9:00 horas. Ao receber as respostas o anfitrião irá detectar uma intersecção e então verificar que é possível alocar esse horário para o compromisso. Os resultados de cada técnica podem ser vistos na tabela 8.

Tabela 8 – Resultados do cenário 1.

Técnica	Iterações	Mensagens de proposta	Mensagens de resposta	Total mensagens
esforço	5	8	8	20
<i>feedback</i>	2	2	2	8
esforço e <i>feedback</i>	2	2	2	8

As quantidades de iterações são contabilizadas após o envio da proposta até a mensagem de confirmação de presença dos envolvidos. Já a coluna que representa a quantidade total de mensagens leva em consideração a quantidade de propostas enviadas para todos os envolvidos, a quantidade de respostas dos envolvidos, o envio de mensagens confirmando o horário e a resposta final de cada envolvido.

Por meio da tabela 8 fica evidente que a técnica que utiliza o esforço para definir o rumo da negociação pode demorar muitas iterações para definir um acordo. Uma possível melhoria nessa técnica seria definir o esforço dinamicamente, de modo que a métrica altere-se conforme as preferências do usuário.

Mesmo apresentando diversas limitações, o seu uso pode ser aplicado junto com outras técnicas, aumentando assim a expressividade e efetividade em negociações.

4.1.2 Cenário 2

Conforme mostrado no cenário 1, a técnica de *feedback* mostrou-se superior em termos de iterações e mensagens trocadas entre os envolvidos em um possível compromisso. Tendo em vista esses pontos, o cenário 2 tem como objetivo destacar uma das limitações de se utilizar apenas o *feedback* para gerar respostas. Para simplificar a apresentação do cenário, vamos considerar os mesmos agentes do cenário anterior, e1, r1 e r2. As agendas de cada agente encontram-se nas tabelas 9, 10 e 11.

Tabela 9 – Agenda de e1.

Data	Dia da semana	Horário	Prioridade	Status
05/09/2016	segunda-feira	7:30	2	disponível
05/09/2016	segunda-feira	8:00	2	disponível
05/09/2016	segunda-feira	8:30	1	disponível
05/09/2016	segunda-feira	9:00	1	disponível
05/09/2016	segunda-feira	9:30	1	disponível
05/09/2016	segunda-feira	10:00	1	disponível
05/09/2016	segunda-feira	10:30	2	disponível

Para a técnica de esforço serão propostos os horários das 8:30, 9:00 e 9:30. Sendo assim, o número máximo de iterações serão 3. Já nas outras duas técnicas o horário inicial será o das 8:30.

Tabela 10 – Agenda de r1.

Data	Dia da semana	Horário	Prioridade	Status
05/09/2016	segunda-feira	7:30	1	disponível
05/09/2016	segunda-feira	8:00	1	indisponível
05/09/2016	segunda-feira	8:30	1	indisponível
05/09/2016	segunda-feira	9:00	1	disponível
05/09/2016	segunda-feira	9:30	1	disponível
05/09/2016	segunda-feira	10:00	2	disponível
05/09/2016	segunda-feira	10:30	2	disponível

Tabela 11 – Agenda de r2.

Data	Dia da semana	Horário	Prioridade	Status
05/09/2016	segunda-feira	7:30	1	disponível
05/09/2016	segunda-feira	8:00	1	disponível
05/09/2016	segunda-feira	8:30	1	indisponível
05/09/2016	segunda-feira	9:00	1	disponível
05/09/2016	segunda-feira	9:30	2	disponível
05/09/2016	segunda-feira	10:00	2	disponível
05/09/2016	segunda-feira	10:30	2	disponível

Ao utilizar a técnica de *feedback* a seguinte situação ocorre: ao receber a proposta, tanto o agente r1, quanto o agente r2 irão buscar em suas agendas e então detectar um conflito com o horário proposto. Após

essa etapa, o agente r1 irá sugerir 7:30 e 9:00 como possíveis horários. O agente r2 irá propor 7:30, 8:00 e 9:00. Ao anexar os horários na resposta, a disposição dos horários sempre será feita de forma crescente.

Após o agente e1 receber as respostas ele irá analisar as respostas e detectar que houve uma intersecção contendo os horários das 7:30 e 9:00. Essa intersecção detectará apenas o horário das 7:30. Entretanto, o horário das 9:00 seria o mais próximo do horário proposto, que conforme mostrado na subseção 3.1.6 apresenta algumas vantagens.

Como a estratégia utilizada no protocolo combina a métrica do esforço e do *feedback*, a quantidade de iterações empregada será maior. A iteração relacionadas à técnica do esforço tem como um dos principais objetivos garantir que mesmo que dois horários sejam detectados na intersecção, o melhor horário, conforme a definição de esforço, seja escolhido. Nesse cenário, o agente e1 irá enviar uma mensagem que contenha os horários das 7:30 e 9:00. Como o horário das 9:00 está mais próximo do horário proposto originalmente, ele será escolhido pelos agentes r1 e r2. Os resultados do cenário 2 podem ser vistos na tabela 12.

Tabela 12 – Resultados do cenário 2.

Técnica	Iterações	Mensagens de proposta	Mensagens de resposta	Total mensagens
esforço	4	3	6	13
<i>feedback</i>	2	2	2	8
esforço e <i>feedback</i>	3	4	4	12

4.1.3 Cenário 3

No cenário utilizado por Hossain e Shakshuki (2013) existe uma hierarquia bem definida, onde alunos estão relacionados com professores e cada membro dessa relação possui um nível. As associações encontram-se na tabela 1. O principal objetivo desse cenário é adaptar a hierarquia de forma que seja possível comparar a expressividade do protocolo desenvolvido nessa trabalho com o trabalho de Hossain e Shakshuki (2013).

Assim como no cenário proposto por Hossain e Shakshuki (2013), o cenário 3 irá iniciar com um convite de reunião feito pelo professor

Darcy, sendo que os convidados são o professor Elhady, dois mestrandos e um graduando. A adaptação da hierarquia usada por Hossain e Shakshuki (2013) estará baseada na prioridade do professor Darcy perante aos estudantes e ao professor Elhady. As prioridades dos envolvidos estão definidas conforme a tabela 13.

Tabela 13 – Relação hierárquica adaptada de Hossain e Shakshuki (2013).

Envolvidos	Darcy	Elhady	Mozamma	Wael	Qianli
Darcy	-	1	2	2	3
Elhady	1	-	2	2	3
Mozamma	3	1	-	2	3
Wael	3	1	2	-	3
Qianli	3	1	2	3	-

Por meio da tabela 13 é possível notar que Darcy possui uma prioridade baixa com os estudantes, porém o professor Elhady possui uma prioridade alta com os alunos. Para tornar o cenário mais direto e simples, as agendas dos alunos terão os horários iguais, conforme mostra a tabela 15. Assume-se que o professor Elhady irá aceitar o convite de Darcy e que os horários disponíveis que serão usados no convite serão os que estão na tabela 14.

Tabela 14 – Agenda do professor Darcy.

Data	Dia da semana	Horário	Prioridade	Status
05/09/2016	segunda-feira	7:30	1	disponível
05/09/2016	segunda-feira	8:00	1	disponível
05/09/2016	segunda-feira	8:30	1	disponível

Tabela 15 – Agenda dos estudantes.

Data	Dia da semana	Horário	Prioridade	Status
05/09/2016	segunda-feira	7:30	2	disponível
05/09/2016	segunda-feira	8:00	2	disponível
05/09/2016	segunda-feira	8:30	2	disponível
05/09/2016	segunda-feira	9:00	1	disponível
05/09/2016	segunda-feira	9:30	1	disponível

Através da adaptação das relações hierárquicas da tabela 13,

nota-se que a prioridade do professor Darcy nas agendas dos alunos é 3 e que em nenhuma das agendas existe um horário com prioridade 3. Ao enviar uma mensagem propondo um horário, por exemplo, segunda-feira, dia 05/09/2016 às 7:30, apenas o professor Elhady iria aceitar o convite.

Conforme mostrada na seção 3.2.1.1, os alunos iriam utilizar a prioridade do grupo para tomada de decisão, obtendo uma média de 2, o que significa que apesar do professor Darcy não ter uma prioridade alta, a prioridade do professor Elhady e dos outros envolvidos iriam ser decisivas na busca por um horário. Sendo assim, o horário proposto por Darcy seria aceito por todos.

Diferente da implementação de Hossain e Shakshuki (2013), o protocolo desenvolvido nesse trabalho mostra que os agentes envolvidos não necessitam ter acesso às respostas dos outros participantes para aceitar ou recusar o convite. O protocolo também leva em consideração que apenas o anfitrião tenha acesso às informações, isso também evita que a agenda dos usuários sejam disponibilizadas à terceiros.

Através dos cenários 1, 2 e 3 é possível notar que existem algumas limitações quando adota-se apenas uma única estratégia perante a situações de negociação. Porém, ao combiná-las é possível obter um aumento em sua expressividade, garantindo que a alocação de compromissos seja mais efetiva.

5 CONCLUSÃO

Neste trabalho foi proposto um protocolo de negociação para SMA que emprega o uso da geração de *feedback* e a métrica do esforço como estratégias de negociação. Esse protocolo tem como objetivo o agendamento automático de reuniões, onde cada agente representa um usuário.

Como agentes representam diferentes usuários, nem sempre eles possuem os mesmos objetivos. Já que um determinado usuário pode preferir ter reuniões pela manhã, enquanto outro prefere ter reuniões apenas em um certo dia da semana. Sendo assim, é preciso que agentes colaborem entre si para que reuniões sejam agendadas.

Ao levar em consideração a região de aceitação de um agente, é possível definir estratégias de negociação para que a cada iteração a proposta chegue mais perto dos pontos definidos por essa região (JENNINGS et al., 2001). Tornando a etapa de negociação fundamental para resolução de conflitos.

Através da utilização das linguagens de programação C++ e Prolog, desenvolveu-se um protótipo do protocolo apresentado nessa pesquisa. O funcionamento do protótipo está baseado em acessos à base de crenças, onde são feitas leituras de prioridades e atualização de disponibilidade de horários. As consultas à base de crenças são realizadas por meio de uma interface entre C++ e Prolog.

Os cenários apresentados na seção 4.1 mostram que a aplicação de apenas uma técnica de negociação pode acarretar em alguns problemas. Ao utilizar apenas a métrica do esforço pode ser gerado um atraso na definição de um consenso, o que pode gerar uma sobrecarga na troca de mensagens. Ao adotar a utilização do *feedback* como resposta, é possível notar casos onde a alocação do compromisso não representa da melhor forma as preferências do usuário.

O protocolo desenvolvido defende o uso de várias técnicas de negociação, de forma que ocorra um aumento na expressividade das regiões de aceitação dos agentes. Os cenários utilizados para testes podem ser uma indicação para comprovar a necessidade do uso de duas ou mais técnicas de negociação em SMA para resolução de conflitos no agendamento automático de reuniões.

5.1 TRABALHOS FUTUROS

Conforme apresentado nos trabalhos da seção 2.6, existem diversas estratégias que melhoram a maneira como agentes negociam. Utilizar informações de reuniões já agendadas para melhorar o modo como as preferências de usuários são representadas pode ser uma maneira de melhorar a definição de prioridades do protocolo proposto. O cancelamento e reagendamento de reuniões, alteração e sobrescrita de prioridades são fundamentais para tornar o protocolo mais robusto e representativo.

A análise matemática do protocolo também é um passo importante para verificar a maximização dos ganhos de todos os agentes envolvidos na negociação.

Outra etapa importante para validar o que foi mostrado nesse trabalho é a integração do protocolo desenvolvido com alguma plataforma de implementação de um SMA. Durante essa etapa de implantação novos testes devem ser realizadas, de modo a definir o quão eficiente e escalável é o protocolo e então definir suas principais limitações.

Como o protocolo apresentado teve seu foco voltado para a definição de estratégias e consultas à base de crenças, fica como trabalho futuro adaptar o protocolo para a inserção dos pontos levantados anteriormente.

REFERÊNCIAS

- AKNINE, S.; ARIB, S.; BOUKREDERA, D. Modeling a multi-issue negotiation protocol for agent extensible negotiations. In: SPRINGER. *European Conference on Multi-Agent Systems*. [S.l.], 2014. p. 349–359.
- AUSTIN, J. L. *How to do things with words*. [S.l.]: Oxford university press, 1975.
- BRATMAN, M. Intention, plans, and practical reason. 1987.
- CAO, M. et al. Automated negotiation for e-commerce decision making: A goal deliberated agent architecture for multi-strategy selection. *Decision Support Systems*, Elsevier, v. 73, p. 1–14, 2015.
- CLOCKSIN, W.; MELLISH, C. S. *Programming in PROLOG*. [S.l.]: Springer Science & Business Media, 2003.
- DASTANI, M. 2apl: a practical agent programming language. *Autonomous agents and multi-agent systems*, Springer, v. 16, n. 3, p. 214–248, 2008.
- GUTTMAN, R. H.; MOUKAS, A. G.; MAES, P. Agent-mediated electronic commerce: a survey. *The Knowledge Engineering Review*, Cambridge Univ Press, v. 13, n. 02, p. 147–159, 1998.
- HOSSAIN, S. M.; SHAKSHUKI, E. A negotiation protocol for meeting scheduling agent. *Procedia Computer Science*, Elsevier, v. 21, p. 164–173, 2013.
- JENNINGS, N. R. et al. Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation*, Springer, v. 10, n. 2, p. 199–215, 2001.
- KRAUS, S. Negotiation and cooperation in multi-agent environments. *Artificial intelligence*, Elsevier, v. 94, n. 1, p. 79–97, 1997.
- LABROU, Y.; FININ, T.; PENG, Y. Agent communication languages: The current landscape. *IEEE Intelligent systems*, IEEE Educational Activities Department, v. 14, n. 2, p. 45–52, 1999.
- MEGASARI, R. et al. Towards host-to-host meeting scheduling negotiation. *International Journal of Advances in Intelligent Informatics*, v. 1, n. 1, p. 23–29, 2015.

MIHAYLOV, M. *Decentralized coordination in multi-agent systems*. Tese (Doutorado) — PhD thesis, Vrije Universiteit Brussel, Brussels, 2012.

PEPPAS, P. Belief revision. *Foundations of Artificial Intelligence*, Elsevier, v. 3, p. 317–359, 2008.

RAHWAN, I. et al. Stratum: A methodology for designing heuristic agent negotiation strategies. *Applied Artificial Intelligence*, Taylor & Francis, v. 21, n. 6, p. 489–527, 2007.

SHAKSHUKI, E. et al. A distributed multi-agent meeting scheduler. *Journal of Computer and System Sciences*, Elsevier, v. 74, n. 2, p. 279–296, 2008.

SIERRA, C. et al. A framework for argumentation-based negotiation. In: *Intelligent agents IV agent theories, architectures, and languages*. [S.l.]: Springer, 1997. p. 177–192.

WIELEMAKER, J. et al. Swi-prolog. *Theory and Practice of Logic Programming*, Cambridge Univ Press, v. 12, n. 1-2, p. 67–96, 2012.

WOOLDRIDGE, M. *An introduction to multiagent systems*. [S.l.]: John Wiley & Sons, 2009.

WOOLDRIDGE, M.; JENNINGS, N. R. et al. Intelligent agents: Theory and practice. *Knowledge engineering review*, Cambridge Univ Press, v. 10, n. 2, p. 115–152, 1995.

WOOLDRIDGE, M. J. *Reasoning about rational agents*. [S.l.]: MIT press, 2000.

WURMAN, P. R.; WELLMAN, M. P.; WALSH, W. E. The michigan internet auctionbot: A configurable auction server for human and software agents. In: *ACM. Proceedings of the second international conference on Autonomous agents*. [S.l.], 1998. p. 301–308.

ZHANG, M. et al. Collaborative agents for complex problems solving. In: *Computational Intelligence*. [S.l.]: Springer, 2009. p. 361–399.

6 ANEXOS

6.1 ANEXO A - *STRATUM*: UMA METODOLOGIA PARA DESENVOLVIMENTO DE ESTRATÉGIAS DE NEGOCIAÇÃO PARA AGENTES

Criado por Rahwan et al. (2007), a metodologia *STRATUM* tem como principal propósito guiar o desenvolvedor de estratégias de negociação, partindo de um domínio e análise de requisitos até a produção de um módulo de alto nível das especificações das estratégias. As especificações devem ser genéricas o suficiente para prover flexibilidade nos detalhes de implementação e ser específica o suficiente para servir de guia durante a implementação das estratégias.

Assim como outras metodologias, a *STRATUM* apresenta uma quantidade de estágios fundamentais para sua utilização. A figura 6 aborda os estágios da metodologia.

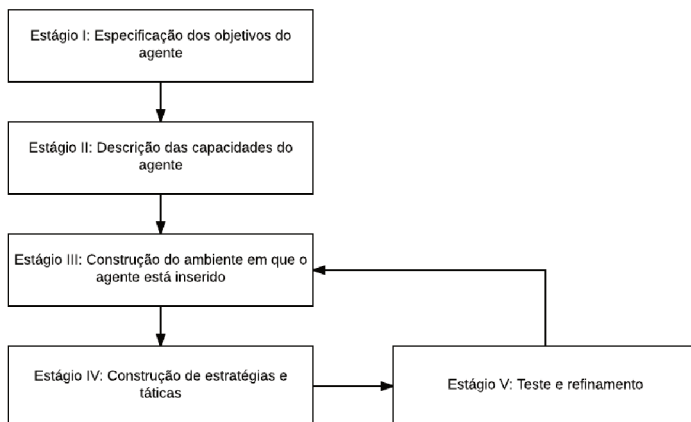


Figura 6 Estágios da metodologia *STRATUM*, adaptado de Rahwan et al. (2007).

Nas próximas subseções serão apresentados as descrições e objetivos de cada estágio da metodologia.

6.1.1 Estágio I: Especificação dos objetivos do agente

A etapa inicial da metodologia descreve os objetivos de cada agente. A complexidade dessa etapa varia sua granularidade em diversos aspectos, ela pode apenas descrever os objetivos do agente de forma geral ou apresentar especificações mais detalhadas dos objetivos e subobjetivos. Também pode ser usada linguagem natural para essa etapa ou alguma especificação formal, utilizando funções de utilidade, por exemplo.

6.1.2 Estágio II: Descrição das capacidades do agente

Durante a etapa de negociação agentes devem realizar uma quantidade específica de ações, sejam elas mentais ou físicas. Sendo assim, essa etapa visa descrever quais as capacidades os agentes presentes na negociação tem disponíveis.

As capacidades de um agente estão fortemente associadas ao protocolo de negociação adotado, tais como: linguagem utilizada e regras do protocolo. Segundo Rahwan et al. (2007), as capacidades podem ser divididas em:

- Habilidades de diálogo: especificação de como o agente irá ser capaz de comunicar-se com outros agentes. Essa capacidade pode ser descrita através dos termos de comunicação definido pelo protocolo de negociação em questão;
- Habilidades relevantes que não envolvem diálogo: esse tipo de capacidade está associado a mecanismos que agentes utilizam durante a comunicação.

6.1.3 Estágio III: Construção do ambiente em que o agente está inserido

Através da especificação de como agentes irão comportar-se durante as interações, esse estágio tem como objetivo prover representações das características do ambiente de negociação. Rahwan et al. (2007) define algumas das principais características da seguintes perguntas:

- O quão agressivo um agente que oferta algo irá comportar-se? E um agente que pretende obter algo?

- O quanto os agentes sabem sobre algo? Quais informações os agentes tem acesso?
- Os agentes irão ser sinceros e confiáveis? Eles irão cumprir com suas promessas?
- Existem padrões que capturam a taxa de mudança do ambiente? Qual a demanda por um determinado recurso? Quais condições afetam as taxas de mudanças?
- Qual será o grau de competitividade dos agentes?

6.1.4 Estágio IV: Construção de estratégias e táticas

Nessa etapa é que são feitas as especificação das táticas e estratégias que serão empregadas durante a negociação. Segundo Rahwan et al. (2007), uma tática pode ser definida como um detalhamento do comportamento e padrões de baixo nível do agente, já estratégias representam comportamentos de negociação mais elaborados e fazem o uso de um conjunto de táticas. Um *template* para a descrição das estratégias e táticas encontra-se na tabela 16

Tabela 16 – Adaptação do modelo de definição de estratégias/táticas do *STRATUM*.

Nome da estratégia	Tática 1
Objetivo	Obter o menor preço de um carro.
Métodos	Informar um preço de um concorrente e realizar outras ofertas
Capacidades	Capacidade para obter informações; Capacidade de impor pressão sobre o vendedor; Capacidade de oferecer nova proposta, aceitar e rejeitar proposta.
Condição de racionalidade	nenhuma
Risco	nenhum

6.1.5 Estágio V: Teste e refinamento

O último estágio da metodologia *STRATUM* pode ser realizar diversas vezes, conforme mostra a figura 6. Os teste podem ser feitos através de análise empírica ou teórica.

Na análise empírica tem-se a execução de várias simulações envolvendo diferentes cenários e estratégias. Os parâmetros de simulações podem ser variados da seguinte forma: diferentes estratégias utilizadas por agentes em uma mesma situação, variação das táticas dentro de uma estratégia, levando em consideração a disponibilidade de informações e as restrições de tempo.

Ao realizar a análise teórica pode-se adotar técnicas utilizadas na teoria dos jogos. O principal objetivo dessas técnicas é a análise do equilíbrio, geralmente analisa-se apenas a implementação dos mecanismos do protocolo e não as estratégias em si.

6.1.6 Considerações finais sobre a metodologia

Rahwan et al. (2007) define casos de estudos voltados para a aplicação da metodologia. Através da figura 6 é possível notar a possibilidade de iterações entre os estágios *III*, *IV* e *V*. O que é realizado na maioria dos ambientes de desenvolvimento, tanto de agentes quanto de estratégias de negociação.

Um dos pontos positivos levantados pelo autor está no fato de que a elaboração e descrição das estratégias são feitas de forma hierárquica, o que pode levar a um melhor entendimento do planejamento das estratégias.

7 ANEXO B - ARTIGO DO TCC

Proposta de um protocolo de negociação em sistemas multiagente para agendamento automático de reuniões

Rodrigo Rodrigues Pires de Mello Thiago Ângelo Gelaim
Ricardo Azambuja Silveira

8 de dezembro de 2016

Resumo

A utilização de um único agente pode gerar um grande custo computacional para atingir um certo objetivo, levando a ineficiência de uso dos recursos. Para tratar tais problemas, o uso de vários agentes torna-se essencial, surgindo a necessidade de desenvolver um sistema multiagente, que tem como um dos principais propósitos atingir os objetivos gerais de um sistema de forma eficiente. Ao inserir diversos agentes em um ambiente, é provável que exista uma disputa pelos recursos limitados ou certas decisões que exigem que os agentes entrem em acordo sobre a próxima ação a ser executada. Para a resolução de conflitos é preciso que técnicas de negociação sejam aplicadas. Tendo em vista esses fatores, esse trabalho propõe um protocolo de negociação em sistemas multiagente para o agendamento automático de reuniões. O desenvolvimento do protótipo do protocolo está baseado na aplicação de duas estratégias de negociação e tem como objetivo aumentar a expressividade dos agentes durante a definição de compromissos.

Palavras-chaves: Agentes. Sistema Multiagente. Negociação. Protocolo. Agendamento Automático de Reuniões.

Introdução

No estudo de agentes um dos principais objetivos é a criação de um sistema que seja autônomo, de forma que quando estiver em um ambiente dinâmico, ele consiga interagir com outras entidades e assim atingir os objetivos do sistema (WOOLDRIDGE, 2009). Atingir um certo objetivo não é o único fator importante. Assim como em outras áreas da computação, o tempo

necessário e o poder computacional para finalizar uma tarefa também estão inerentes ao desenvolvimento de agentes.

Frequentemente requisita-se ajuda de terceiros que necessitam ter alguma espécie de conhecimento para fornecer algum auxílio (WOOLDRIDGE, 2009). Problemas que envolvem colaboração não são exclusivos de sistemas multiagente. Pessoas interagem e colaboram entre si para a resolução de seus objetivos.

Tendo em vista o que foi abordado anteriormente, fica explícita a necessidade do uso de vários agentes. A área que engloba esses aspectos é conhecida como sistemas multiagente, sendo esse um dos tópicos de estudo desse trabalho. Outro ponto importante é a maneira como agentes realizam essas tarefas, quais ações cada um irá executar, como elas serão feitas e em qual ordem, ou seja, como serão definidas as ações de cada agente, de forma a tratar a maneira como eles interagem em sociedade (WOOLDRIDGE, 2009).

A existência de recursos limitados em um ambiente leva a ocorrência de disputas e conflitos entre agentes, tornando essencial a existência de algum mecanismo para a resolução desses problemas (JENNINGS et al., 2001). Para contornar conflitos e disputas é necessário empregar técnicas de negociação, que determinam a distribuição e uso de recursos em um ambiente (JENNINGS et al., 2001). Uma das propostas desse trabalho é o desenvolvimento de um protocolo de negociação para sistemas multiagente, onde cada agente tem como objetivo agendar reuniões em nome do usuário representado por ele.

1 Agendamento automático de reuniões

Em agendamento automático de reuniões um agente representa um usuário e comunica-se com outros agentes para definir um horário para compromissos (HOSSAIN; SHAKSHUKI, 2013). Nesse trabalho será assumido que um compromisso é uma atividade que envolve no mínimo duas pessoas e tem um assunto a ser tratado, ou seja, pode-se definir o compromisso como uma reunião.

A seção 2 apresenta os principais aspectos empregados no desenvolvimento do protocolo. Na seção 3 é mostrado através de pseudo-algoritmos o funcionamento do protocolo e a maneira como as alocações de reuniões são realizadas. Já no capítulo ?? são abordadas decisões de implementação e os cenários utilizados para testar o protótipo do protocolo.

2 Proposta de um protocolo de negociação em sistemas multiagente

Nas próximas seções serão apresentadas as principais etapas utilizadas para a definição e implementação de um protocolo de negociação para o agendamento automático de reuniões. Como forma de descrição do protocolo,

a metodologia utilizada foi a *STRATUM*, criado por [Rahwan et al. \(2007\)](#). Essa metodologia está baseada nas seguintes etapas:

1. Especificação dos objetivos do agente;
2. Descrição das capacidades do agente;
3. Construção do ambiente em que o agente está inserido;
4. Construção de estratégias e táticas;
5. Teste e refinamento.

2.1 Especificação dos objetivos do agente

Os objetivos do agente em agendamento automático de compromissos podem ser divididos em:

1. Alocar uma reunião;
2. Alocar uma reunião em um horário que satisfaça as preferências do usuário;
3. Sua proposta de reunião deve ser aceita pela maior quantidade possível de envolvidos. Ou seja, o agente deve ser flexível a ponto de analisar respostas e definir o melhor horário para o grupo.

2.2 Descrição das capacidades do agente

Para entrar em uma negociação todo agente deve ser capaz de interagir com outros agentes que estão disputando um mesmo recurso ([JENNINGS et al., 2001](#)). A etapa de descrição das capacidades do agente está voltada para a definição de quais capacidades o agente utiliza durante a definição de compromissos ([RAHWAN et al., 2007](#)). A tabela 1 descreve as capacidades de um agente em um AAR (Agendamento Automático de Reuniões).

Tabela 1 – Definição das capacidades dos agentes retirada de (RAHWAN et al., 2007).

	Capacidade	Descrição
C1	Realizar proposta	Definição de um horário, assunto e agentes envolvidos para criar uma proposta.
C2	Aceitar proposta	-
C3	Rejeitar proposta	-
C4	Criar <i>feedback</i>	Criação de uma resposta, que contenha os horários disponíveis para aquele compromisso.
C5	Definir esforço	Todo horário de uma proposta pode ser associado a um valor numérico, que representa o esforço necessário para alocar esse horário.
C6	Criar resposta	Definir uma resposta, utilizando C4 e C5.
C7	Analisar <i>feedback</i>	Ao receber uma resposta com um compromisso, o anfitrião analisa cada uma, de forma a definir o melhor horário em relação ao conjunto de respostas.
C8	Analisar respostas que contenham o esforço	Anfitrião analisa cada resposta, e armazena a soma do esforço e o horário daquela rodada.
C9	Acessar agenda	Buscar e atualizar informações sobre horários de compromissos.
C10	Acessar preferências do usuário	Buscar informações relacionadas às preferências do usuário.

2.3 Base de crenças

O funcionamento do protocolo encontra-se na consulta à base de crenças de um agente, esse banco de crenças contém informações relevantes sobre as preferências do usuário que é representado pelo agente, tais como:

- prioridade de envolvidos: quais são as prioridades das pessoas ou organizações que pretende-se agendar um compromisso. Um professor tende a atribuir uma prioridade maior a um orientando do que a um aluno que não esteja envolvido em alguma pesquisa que esteja ligado a ele;
- prioridade de assuntos: semelhante a prioridade de envolvidos, onde um assunto possui um valor que represente sua prioridade na agenda;
- prioridade de horários: definição de quais são os horários em que se deseja agendar um compromisso;

- horários disponíveis e indisponíveis;
- envolvidos em um assunto.

Toda prioridade é definida por valores inteiros que estão entre 1 e 10, onde o valor 1 representa o valor mais prioritário e o valor 10 representa o valor com menor prioridade. A definição desses valores é uma tentativa de reproduzir as preferências do usuário, ou seja, se um usuário deseja ter suas reuniões marcadas sempre no período da manhã, então é necessário que o agente tenha os horários da parte da manhã com a prioridade contendo o valor 1. O mesmo vale para a prioridade de pessoas e assuntos.

2.4 Resposta através do *feedback*

No protocolo proposto essa tarefa pode ser feita através de uma resposta que contenha os horários em que o agente esteja disponível. Mandar uma mensagem com todos os horários disponíveis poderia causar uma sobrecarga na troca de mensagens, então é fundamental utilizar algum mecanismo que seja capaz de identificar quais são os horários apropriados para um determinado compromisso.

Para contornar os problemas identificados anteriormente, é necessário usar as informações que representam as preferências do usuário para estipular os melhores horários. A geração do *feedback* visa associar os horários disponíveis com as informações referentes a prioridade do assunto, do anfitrião (agente que enviou a proposta de reunião) e dos envolvidos. Ao receber uma mensagem enviada por um agente com prioridade 1, todos os horários disponíveis com prioridade 1 serão anexadas na resposta.

2.5 Definição de esforço para alocação de um compromisso

O uso de leilões em sistemas multiagente é um mecanismo muito utilizado para decidir qual agente irá utilizar um certo recurso. Apoiando-se nesses mecanismos e suas derivações, conforme mostrado na seção 2, o protocolo proposto define uma métrica para determinar o esforço que seria necessário para alocar um horário. Essa etapa de definição da métrica ocorre na análise de uma proposta e sua definição pode ser obtida por meio dos seguintes valores:

- ps é o valor da prioridade do assunto;
- pr é o valor da prioridade do anfitrião;
- pg é a média da prioridade dos envolvidos;
- h horário proposto;
- O valor numérico que representa a rodada em que se encontra a negociação também tem influência na definição da métrica de esforço;

A métrica do esforço pode ser definida como: $e = ps + pr + pg + n$. Onde n é o valor normalizado da menor diferença entre o horário proposto e um horário disponível, variando de 0 à 100.

Ao receber um conjunto de respostas o anfitrião leva em consideração a prioridade dos envolvidos combinada com o valor do esforço, dando preferência aos envolvidos que representam uma maior importância para o anfitrião. Todas essas abordagens servem para aproximar as áreas de aceitação dos envolvidos.

3 Estratégias e funcionamento do protocolo

A etapa 4 do *STRATUM* visa definir as estratégias utilizadas na negociação. As próximas subseções apresentam as estratégias adotadas na etapa de negociação e também como o anfitrião analisa cada resposta dos envolvidos, de forma a escolher um horário para a reunião.

Todas as estratégias utilizadas durante a execução do protocolo assumem que o agente está sendo sincero sobre as informações referentes à representação da sua região de aceitação.

3.1 Propondo compromissos

A proposta de uma reunião ocorre quando existe um assunto na base de crenças que ainda não foi tratado em nenhuma reunião. Ao assumir a existência dessa característica, então pode-se considerar que o agente tem a intenção de marcar uma reunião. Tendo disponível o assunto, é necessário estabelecer os envolvidos e os horários que estão associados a esse assunto. Essa associação é feita por meio do valor da prioridade do assunto, com isso consulta-se à base de crença para obter o horário que possui prioridade igual à prioridade do assunto.

3.1.1 Definindo a resposta

Ao receber a proposta de um compromisso serão obtidas informações das prioridades de quem enviou o convite, de quem possivelmente estará na reunião, o assunto e o horário proposto. A geração da resposta pode ser dividida nos seguintes passos:

1. verificar a disponibilidade do horário proposto: caso o horário proposto esteja disponível, então uma mensagem de aceitação é enviada;
2. definir a prioridade do compromisso: busca-se na base de crenças a prioridade do anfitrião;
3. em casos onde o anfitrião represente um usuário com prioridade baixa, o protocolo leva em consideração a prioridade do grupo. A prioridade do grupo é a média aritmética da prioridade de todos os envolvidos no compromisso;

4. Uma consulta à base de crenças utilizará a prioridade do anfitrião ou a prioridade do grupo para obter os horários com prioridade igual;
5. Casos em que nenhum horário tenha retornado da consulta à base de crenças pode demonstrar que os envolvidos ou o anfitrião não possuam relevância suficiente, porém o assunto pode ser um fator importante para o usuário. Então uma nova consulta é realizada, porém levando em consideração a prioridade do assunto;
6. Uma mensagem com todos os possíveis horários em que seja possível alocar a reunião é enviada;
7. Caso nenhum horário seja encontrado, uma mensagem recusando o convite será enviada.

Mesmo negando um convite o agente ainda deve participar das próximas iterações da negociação, de forma que ele ainda tenha chances de receber um horário que satisfaça suas condições de agendamento.

3.2 Analisando as respostas

Ao receber as respostas da sua proposta o anfitrião deve procurar por um horário que seja aceito por todos os envolvidos. Como nem sempre isso é possível, existem duas estratégias que podem ser adotadas: procurar pelos horários que sejam aceitos pelo maior número de envolvidos ou o horário que seja aceito pelos envolvidos que possuem maior prioridade.

Quando um horário em comum é encontrado, apenas envia-se uma mensagem confirmando o horário e então atualiza-se a base de crenças do anfitrião. Já em casos onde encontrou-se n horários que foram obtidos a partir da procura citada acima, esses horários são anexados em uma mensagem e então enviados novamente para os envolvidos.

3.3 Lidando com impasses e conjunto de horários

Ao receberem a nova mensagem com o conjunto de horários cada envolvido deve definir o esforço para o horário contido na mensagem e então enviar a resposta com esse valor. É válido ressaltar que o esforço analisa a distância entre o horário proposto e os horários que possuem prioridade próximas ao compromisso. Com isso tem-se o horário alternativo mais próximo do horário proposto.

Quando o anfitrião receber essas respostas ele deve procurar pelo horário que representa o menor esforço. Essa procura pela menor soma de esforço serve tanto para casos em que nenhum horário foi definido, quanto para existência de dois ou mais horários aceitos por todos os envolvidos. A negociação pode ser finalizada em dois casos:

- Quando o anfitrião receber as respostas com o *feedback*, ele detecta um horário comum entre os participantes;
- A quantidade máximas de rodadas foi atingida e então obtém-se o horário que representa o menor esforço de todos os participantes.

4 Cenário e testes realizados

Para testar o protocolo apresentado nas seções anteriores foi definido um cenário, que tem como objetivo comparar as técnicas de negociação. Nesse cenário serão comparadas a métrica do esforço, a geração do *feedback* e uma combinação da métrica do esforço com o *feedback*. É válido ressaltar que a combinação de ambas as técnicas formam a estratégia utilizada pelos agentes durante a execução do protocolo de negociação.

4.1 Cenário 1

O cenário 1 envolve três agentes que serão identificados como e1, r1 e r2. Sendo que o agente e1 irá enviar uma mensagem propondo um horário para uma reunião, já os agentes r1 e r2 irão interagir com o e1 para tentar definir um horário em comum. Um trecho da agenda de cada um dos agentes envolvidos podem ser vistas nas tabelas 3 e 4.

Nesse cenário será utilizado um assunto que possui prioridade 1, sendo esse o assunto não resolvido com maior prioridade. Ao utilizar apenas o esforço como estratégia, os horários propostos a cada iteração serão os seguintes: 7:30, 8:00, 8:30 e 9:00.

Tabela 2 – Trechos das agendas de e1, r1 e r2.

Tabela 3 – Trecho da agenda de e1.

Horário	Prioridade	Status
7:30	1	disponível
8:00	1	disponível
8:30	1	disponível
9:00	1	disponível
9:30	2	disponível
10:00	2	disponível
10:30	2	disponível

Tabela 4 – Trecho da agenda de r1 e r2.

Horário	Prioridade	Status
7:30	1	indisponível
8:00	1	indisponível
8:30	1	indisponível
9:00	1	disponível
9:30	2	disponível
10:00	2	disponível
10:30	2	disponível

Ao utilizar a métrica do esforço é possível concluir que o único horário possível para o compromisso é o das 9:00, que será proposto na última iteração da negociação, levando 5 rodadas para chegar em um acordo. Isso evidencia algumas limitações que foram descritas na subseção 2.5.

Já nas outras duas abordagens é possível notar que o acordo ocorrerá em apenas uma iteração. Os resultados de cada técnica podem ser vistos na tabela 5.

Tabela 5 – Resultados do cenário 1.

Técnica	Iterações	Mensagens de proposta	Mensagens de resposta	Total mensagens
esforço	5	8	8	20
<i>feedback</i>	2	2	2	8
esforço e <i>feedback</i>	2	2	2	8

Por meio da tabela 5 fica evidente que a técnica que utiliza o esforço para definir o rumo da negociação pode demorar muitas iterações para definir um acordo. Uma possível melhoria nessa técnica seria definir o esforço dinamicamente, de modo que a métrica altere-se conforme as preferências do usuário.

Conclusões

Neste trabalho foi proposto um protocolo de negociação para SMA que emprega o uso da geração de *feedback* e a métrica do esforço como estratégias de negociação. Esse protocolo tem como objetivo o agendamento automático de reuniões, onde cada agente representa um usuário.

Ao levar em consideração a região de aceitação de um agente, é possível definir estratégias de negociação para que a cada iteração a proposta chegue mais perto dos pontos definidos por essa região (JENNINGS et al., 2001). Tornando a etapa de negociação fundamental para resolução de conflitos.

O cenário apresentado na seção 4 mostra que a aplicação de apenas uma técnica de negociação pode acarretar em alguns problemas. Ao utilizar apenas a métrica do esforço pode ser gerado um atraso na definição de um consenso, o que pode gerar uma sobrecarga na troca de mensagens. Ao adotar a utilização do *feedback* como resposta, é possível notar casos onde a alocação do compromisso não representa da melhor forma as preferências do usuário.

O protocolo desenvolvido defende o uso de várias técnicas de negociação, de forma que ocorra um aumento na expressividade das regiões de aceitação dos agentes. Um dos cenários utilizados para testes podem ser uma indicação para comprovar a necessidade do uso de duas ou mais técnicas de negociação em SMA para resolução de conflitos no agendamento automático de reuniões.

5 Trabalhos futuros

A análise matemática do protocolo também é um passo importante para verificar a maximização dos ganhos de todos os agentes envolvidos na negociação.

Outra etapa importante para validar o que foi mostrado nesse trabalho é a integração do protocolo desenvolvido com alguma plataforma de implementação de um SMA. Durante essa etapa de implantação novos testes devem ser realizadas, de modo a definir o quão eficiente e escalável é o protocolo e então definir suas principais limitações.

Como o protocolo apresentado teve seu foco voltado para a definição de estratégias e consultas à base de crenças, fica como trabalho futuro adaptar o protocolo para a inserção dos pontos levantados anteriormente.

Referências

- HOSSAIN, S. M.; SHAKSHUKI, E. A negotiation protocol for meeting scheduling agent. *Procedia Computer Science*, Elsevier, v. 21, p. 164–173, 2013. Citado na página 2.
- JENNINGS, N. R. et al. Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation*, Springer, v. 10, n. 2, p. 199–215, 2001. Citado 3 vezes nas páginas 2, 3 e 9.
- RAHWAN, I. et al. Stratum: A methodology for designing heuristic agent negotiation strategies. *Applied Artificial Intelligence*, Taylor & Francis, v. 21, n. 6, p. 489–527, 2007. Citado 2 vezes nas páginas 3 e 4.
- WOOLDRIDGE, M. *An introduction to multiagent systems*. [S.l.]: John Wiley & Sons, 2009. Citado 2 vezes nas páginas 1 e 2.